# Transmission Scheduling with Deadline and Throughput Constraints

by

Kyu Seob Kim

B.S. Electrical and Computer Engineering, Cornell University (2011)

and

B.A. Mathematics, Cornell University (2011)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013





Author ..........................................................................
Department of Electrical Engineering and Computer Science
July 16, 2013

Certified by..........................
Eytan Modiano
Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by..........................
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students
Department of Electrical Engineering and Computer Science

# Transmission Scheduling with Deadline and Throughput Constraints

by

Kyu Seob Kim

Submitted to the Department of Electrical Engineering and Computer Science
on July 16, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science

## Abstract

Today wireless networks are increasingly used to perform applications with Quality of Service constraints (QoS) such as delay, delivery ratio, and channel reliability. Especially, as demand for real-time transmissions increases, transmitting packets with hard delivery deadlines poses an important network control problem. In this thesis, we propose a framework for characterizing feasibility regions and finding an optimal scheduling policy in various wireless networks. First, we start with a wireless network with multiple unicast flows. We investigate how delay in feedback information decreases the feasibility region. Second, we consider time-varying channels and how delay in network state information decreases the feasibility region. Third, we characterize the feasibility region of a wireless network with multiple multicast flows. In each case, we characterize the feasibility region, prove that a max-weight policy is a feasibility optimal policy and present the results of simulation studies verifying the theoretical studies.

Thesis Supervisor: Eytan Modiano
Title: Professor, Department of Aeronautics and Astronautics

## Acknowledgments

Thank you to my adviser, Prof. Modiano, for all of his wisdom and advice.

Thank you to Dr. Chih-ping Li for proofreading my thesis.

Kyu Seob Kim

Cambridge, MA

July 16, 2013

THIS PAGE INTENTIONALLY LEFT BLANK

# Contents

# List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 1

# Introduction

Wireless networks provide a flexible platform to support a variety of applications such as voice, multimedia, data, and messaging. With more and more clients using wireless technologies and mobile devices, demand for real-time transmissions of audio and visual content has increased significantly. Hence, how to efficiently transmit real-time data over wireless channels is becoming an important network control problem.

It is difficult to reliably transmit real-time data over wireless channels. First, unlike elastic traffic such as e-mail, which can be supported by best-effort services, real-time data has strict delay constraints. Second, in a wireless network, strict delay constraints are hard to satisfy because wireless channels are unreliable and the channel states may vary over time. Data packets can get lost and may need to be retransmitted, which causes additional delay.

In some of the related works, researchers have focused on the network control problem of serving real-time data packets in a unicast system [2,4,5]. Hou et al developed a novel analytical framework of serving packets with hard deadlines over unreliable channels in [2]. They consider a wireless network consisting of a base station and $N$ wireless clients and group $T$ time slots into a frame to model the delay constraints. One packet for each client arrives at the beginning of the frame and all the undelivered packets at the end of the frame are dropped. As a part of the QoS (Quality of Service) requirement, every client has a delivery ratio requirement, i.e. the long-term ratio of the packets delivered successfully. Then, the feasibility region can be defined as the set of delivery ratio requirement vectors such that there exists a policy that fulfills them and a feasibility optimal policy a policy that can fulfill any delivery ratio requirement vectors that are inside the feasibility region. In [2], Hou et al assume instant feedback, characterize the feasibility region, and suggest two debt-based feasibility optimal policies called the *largest time-based debt first policy* and the *largest weighted-delivery debt first policy.* In [5], Hou and Kumar study probabilistic packet arrivals instead of assuming that there is one packet for every client at the beginning of the frame. In [4], they extend the model further to incorporate time-varying channels, where the channel condition changes according to an ergodic and irreducible Markov chain with a finite number of states. They characterize the feasibility region and develop a feasibility optimal policy by decoupling the channel states. In addition to characterizing the feasibility region and feasibility optimal policies, Hou and Kumar studied the utility maximization problem by considering the delivery ratio requirements as changeable parameters for static wireless networks [6].

Depending on the size and the structure of the network, however, instant feedback may not be available or difficult to obtain. For example, in a typical satellite communications system, the delay can be more than 100ms. Even in a regular cell phone network, depending on the distance between a client and the base station, the feedback delay can be significant. In Chapter 3, we strengthen the model proposed in [2] to incorporate delayed feedback. We analyze how delayed feedback changes the feasibility region and develop a feasibility optimal policy under delayed feedback by formulating and solving a dynamic program.

In Chapter 3, we assume static channels. However, in real world, wireless channel state vary over time due to node mobility, fading, or scattering. Hou and Kumar modeled the time-varying wireless channels using an ergodic and irreducible Markov chain with a finite number of states in [4]. Assuming that the current channel state information is available, they characterize the feasibility region and a feasibility optimal policy by decouping the channel states.

The current channel state may not be available or difficult to obtain in certain networks. For example, in some communications system, the current channel state is estimated from the past delivery results. Hence, in Chapter 4, we strengthen the model proposed in [4] to incorporate delayed channel state information. Instead of assuming that the current channel state information is available, we assume that the channel state information is delayed. We examine how the delay in the channel state information affects the feasibility region and develop feasibility optimal policies. We also extend the utility maximization frame developed in [6] to include the time-varying channel case.

In Chapter 3 and Chapter 4, only unicast traffic has been considered, but some real-time data traffic is multicast in nature and should be treated differently from the unicast traffic. For example, in the real time broadcast system, different clients subscribe to the same flow and receive the same data packets. Unlike the unicast traffic, different clients that subscribe to the same flow may have different wireless channel qualities. Consider an example wireless network in which there are two flows.

In [3], Hou and Kumar study multicast flows without feedback and develop a feasibility optimal policy by solving a maximization problem. Assuming no feedback is appropriate to analyze a large wireless network since instant feedback can be very difficult and impractical to obtain in such a large network. However, in a network of moderate size, instant feedback can be readily available and a scheduling policy can utilize such information. In Chapter 5, we focus on a wireless network of moderate size and assume that the instant feedback is always available. We characterize the feasibility region, which is the set of all the delivery ratio requirement vectors that can be supported by any scheduling policy and develop a feasibility optimal policy, which can satisfy any delivery ratio requirement vectors that are within the feasibility region.

THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 2

# Model and Preliminary Results

The wireless network with unicast flows with instant feedback has been studied extensively by Hou et al in [2]. They first develop necessary conditions for the feasibility region, suggest two *largest debt first* policies, prove that they are feasibility optimal, and finally prove that the necessary conditions are sufficient as well by showing that the feasibiltiy optimal policies fulfill all delivery ratio requirement vectors that satisfiy the necessary conditions. In this chapter, we provide the system model, review background material, and develop a numerical method to characterize the feasibility region. This will be useful in determining the feasibility regions for more general cases later such as unicast with delayed feedback and multicast because the closed-form characterization is difficult to obtain in such cases. Also the numerical method plays a crucial role in developing the feasibility optimal policy in the more general cases.

**Definition 2.0.1** The unicast system is a system in which a flow is subscribed to by only one client and one client subscribes to only one flow.

We explain our system model in the following section.

## 2.1 System Model

Consider a wireless system where there is one base station sending data packets with delay constraints to $N$ wireless clients. This model is the same as the model used in [2].

Since the link between the server and a client is wireless, we assume that client $n$ has a channel reliability of $p_n$, i.e. when the server transmits a packet to client $n$, the probability of successful transmission is $p_n$. The value of $p_n$ depends on the client and the different values reflect that wireless links can vary in quality from client to client.

We consider a time-slotted system and group $T$ time slots into a frame as shown in Figure 2-1. All packets arrive at the beginning of the frame, and the base station transmits a packet in each slot. Note that since the wireless channel is unreliable, not all packets may be delivered to their destination.

Define the delay of a packet as the number of time slots between the arrival of the packet to the base station and the end of the time slot when the successful transmission happens. For example, if packet 1 is successfully delivered during time slot 0 in Figure 2-1, then the delay of packet 1 is one time slot. To model the hard deadline constraints, we assure that the delay of each delivered packet is less than $T$ by dropping any undelivered packets at the end of a frame as shown in Figure 2-1.

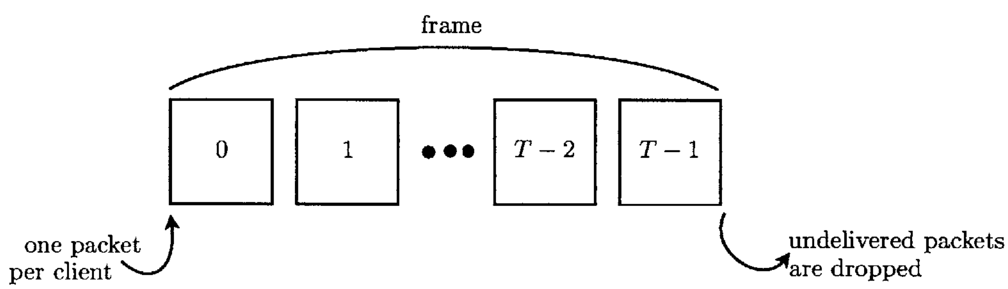



Figure 2-1: $T$ time slots constitute a frame.

A time slot is long enough for a transmission of a packet to occur. Thus, if a packet transmission is scheduled during a time slot and is successful, then the packet is delivered by the end of the time slot. The base station receives an ACK/NACK signal by the end of the time slot during which a packet is served, i.e. the instant feedback is available.

**Definition 2.1.1** (Definition 4 in [2]) A scheduling policy is *work conserving* if it never idles whenever there is an undelivered packet in the system

**Definition 2.1.2** A scheduling policy is *non-anticipatory* if it does not use future information

We consider the class of work-conserving and non-anticipatory scheduling policies and denote the class by $\Pi$. The performance measure we are interested in is the long-term proportion of packets delivered. Let $D_i^\eta(k)$ be the indicator random variable that is equal to 1 if client $i$ receives the packet during the interval $[kT, (k+1)T)$ by following policy $\eta$ and 0, otherwise. Then, the long-term proportion $\tilde{q}_i^\eta$ of successful deliveries of packets to client $i$ can be written as:

$$\tilde{q}_i^\eta := \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} D_i^\eta(k)$$

Let $\mathcal{H}_{k-1}^\eta$ denote the history of all packet deliveries up to and including frame $k-1$ under policy $\eta$. That is, $\mathcal{H}_{k-1}^\eta$ is a vector $(\vec{D}^\eta(0),\cdots,\vec{D}^\eta(k-1))$ of indicator variable vectors $\vec{D}^\eta(t) = (D_i^\eta(t))_{i=1}^N$. Then, we define the expected long-term throughput $\hat{q}_i^\eta$ for client $i$ as

$$\begin{aligned}\hat{q}_i^\eta :&= \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} E[D_i^\eta(k)|\mathcal{H}_{k-1}^\eta] \\ &= \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} E[D_i^\eta(k)|\vec{D}^\eta(0),\cdots,\vec{D}^\eta(k-1)]. \end{aligned} \tag{2.1}$$

Let $(X_k)$ be a sequence of random variables and $(b_k)$ be a sequence of real numbers that is monotonically increasing to $\infty$. Then, we have the following theorem:

**Theorem 2.1.1** (Stability Theorem in [8]) If $\sum_k \frac{\mathrm{Var}(X_k)}{b_k^2} < \infty$ with $b_k$ monotonically increasing to $\infty$, then

$$\frac{1}{b_{K-1}}\sum_{k=0}^{K-1}\{X_k - E(X_k|X_0,\cdots,X_{k-1})\} \xrightarrow{\text{a.s.}} 0.$$

By setting $b_k = k$ and applying the above theorem, we can show that $\tilde{q}_i^\eta = \hat{q}_i^\eta$ with probability 1. In this thesis, we use $\hat{q}_i^\eta$ exclusively to characterize the feasibility region and the feasibility optimal policy.

As part of the QoS constraints, each client $i$ has a specified delivery ratio requirement $q_i$, i.e. the long-term proportion of expired packets cannot exceed $1 - q_i$.

**Definition 2.1.3** Delivery ratio requirement vector $(q_n)_{n\in\{1,\cdots,N\}}$ is said to be fulfilled by policy $\eta \in \Pi$ if $\hat{q}_n^\eta \geq q_n$ for every client $n$ with probability 1.

**Definition 2.1.4** The feasibility region $\Lambda$ is the set of delivery ratio requirement vectors such that there exists a policy $\eta \in \Pi$ that fulfills the vectors.

Once we characterize the feasibility region, we can think of a policy that fulfills all the delivery ratio requirement vectors in the feasibility region.

**Definition 2.1.5** (Definition 3 in [2]) A scheduling policy is said to be feasibility optimal if it fulfills every delivery ratio requirement vector in the feasibility region $\Lambda$.

## 2.2 Feasibility Region

Although Hou et al analyze the feasibility region in [2], here we provide a numerical method to characterize the feasibility region.

For a scheduling policy $\eta \in \Pi$, let $D_i^\eta(k)$ be a random variable that is equal to 1 if a packet is successfully transmitted to client $i$ during frame $k$ by following policy $\eta$ and 0 otherwise. To find a policy that maximizes the long-term expected weighted sum throughput, it is sufficient to consider a policy over only one frame because the packets arrive periodically at the beginning of each frame and any undelivered packets at the end of the frame are dropped. Hence, from now on, we only consider the first frame. For a given weight vector $(\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, the expected weighted sum throughput of $\eta$ during the first frame is $\sum_{i=1}^N E[\alpha_i D_i^\eta(0)]$.

**Definition 2.2.1** For a given weight vector $\vec{\alpha} = (\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, we define the maximum expected weighted sum throughput during one frame as

$$\text{EWST}(\vec{\alpha}) := \max_{\eta \in \Pi} \sum_{i=1}^{N} E[\alpha_i D_i^\eta(0)]$$

**Lemma 2.2.1** Let $\Lambda$ denote the feasibility region. Let $\vec{q}$ represent a feasible delivery ratio requirement vector, i.e. $\vec{q} \in \Lambda$. Then, for any weight vector $\vec{\alpha} = (\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, $\vec{\alpha} \cdot \vec{q} \leq \text{EWST}(\vec{\alpha})$

*Proof.* By definition, $\vec{q} \in \Lambda$ implies that there exists a scheduling policy $\pi \in \Pi$ such that

$$\liminf_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} E[D_i^\pi(k) | \mathcal{H}_{k-1}^\pi] \geq q_i, \quad \text{w.p.1} \quad \forall i \in \{1, 2, \cdots, N\}$$

Because $\text{EWST}(\vec{\alpha})$ is the maximum expected weighted sum throughput over one frame, $\text{EWST}(\vec{\alpha})$ is larger than or equal to $\sum_{i=1}^N E[\alpha_i D_i^\pi(k) | \mathcal{H}_{k-1}^\pi]$ for any $k$ with probability 1. Therefore,

$$\text{EWST}(\vec{\alpha}) \geq \sum_i \alpha_i q_i = \vec{\alpha} \cdot \vec{q}. \qquad \square$$

From Lemma 2.2.1, we know that a hyperplane defined by $\{\vec{y} | \vec{\alpha} \cdot \vec{y} = \text{EWST}(\vec{\alpha})\}$ is tangent to the boundary of the feasibility region as described in Figure 2-2. Such a hyperplane is referred to as a maximizing hyperplane. Then, by varying the weight vector $(\alpha_i)$ and finding the intersections of the maximizing hyperplanes, we can characterize the boundary of the feasibility region numerically.

We can interpret the expected weighted sum throughput as reward $\alpha_i$ is awarded whenever

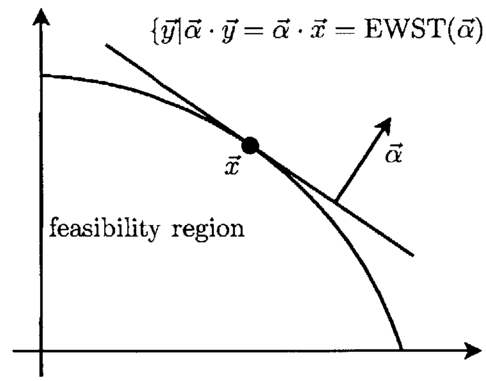



Figure 2-2: Maximizing hyperplanes characterize the outermost boundary of the feasibility region.

the packet transmission to client $i$ is successful. Then, we can think of a policy that consists of a sequence of functions that maximize the expected immediate reward.

**Definition 2.2.2** The greedy policy is a policy that consists of a sequence of functions that maximize the expected immediate reward.

**Example 2.2.1** Let $\mathcal{N}(t)$ denote the set of clients who have not received packets yet at the beginning of time slot $t$. Then, serving client $i$ during time slot $t$ yields the expected immediate reward of $\alpha_i p_i$ if $i \in \mathcal{N}(t)$ and 0 otherwise. Let $\pi_{\text{greedy}} = \{\mu_0, \cdots, \mu_{T-1}\}$ represent the greedy policy. Then, $\mu_t(\mathcal{N}(t)) = \arg\max_{i\in\mathcal{N}(t)} \alpha_i p_i$.

Next, we prove that the greedy policy achieves $\mathrm{EWST}(\vec{\alpha})$.

**Theorem 2.2.1** Consider a unicast system with instant feedback. Let $\pi_{\text{greedy}}$ denote the greedy policy. Then, for any given weight vector $\vec{\alpha}$ such that $\alpha_i \geq 0$, $\mathrm{EWST}(\vec{\alpha}) = \sum_{i=1}^{N} E[\alpha_i D_i^{\pi_{\text{greedy}}}(0)]$.

*Proof.* See Appendix A $\square$

Using the policy in Theorem 2.2.1, we draw the maximizing hyperplanes while varying the weight vectors in Figure 2-3.

Unlike the numerical method presented above, Hou et al derive a closed-form expression for the unicast feasibility region in [2]. Let $p_n$ denote the channel reliability for client $n$, $q_n$ denote the required delivery ratio for client $n$ and $T$ denote the number of time slots per frame.

**Lemma 2.2.2** (Lemma 1 in [2]) The delivery ratio of client $n$ is at least $q_n$ with probability 1 if and only if the long-term time average of the proportion of time slots during which the base station is transmitting to client $n$ is at least $w_n = \frac{q_n}{p_n T}$.

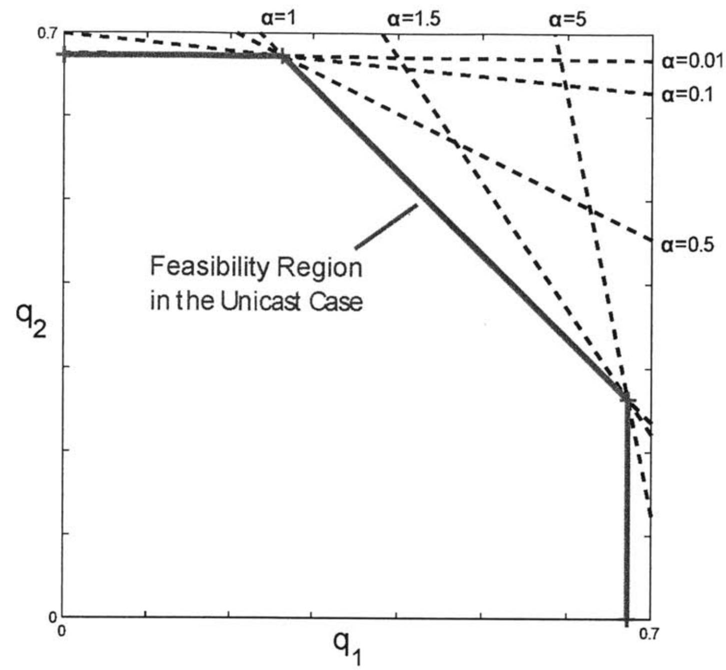



Figure 2-3: The hyperplanes maximizing the weighted expected sum throughput coincide with the feasibility region in [2] ; $p_1 = p_2 = 0.2$, $T = 5$, and we maximize $E\left[\alpha D_1(0) + D_2(0)\right]$ with $(\alpha, 1)$ being the weight vector $(\alpha \geq 0)$.

Using Lemma 2.2.2, the feasible set of clients can be obtained.

**Theorem 2.2.2** (Theorem 4 in [2]) A set of clients is feasible if and only if $\sum_{n \in L} w_n \leq 1 - I_L$ for every subset $L$ of the clients, where $I_L := \frac{E[\max\{0, T - \sum_{n \in L} \gamma_n\}]}{T}$, $\gamma_n$ is geometrically distributed R.V. with parameter $p_n$ and $w_n = \frac{q_n}{p_n T}$.

The hyperplanes maximizing the weighted expected sum throughput coincide with the result from Theorem 2.2.2 as shown in Figure 2-3.

## 2.3 Feasibility Optimal Policy

After we characterize the feasibility region, we develop a feasibility optimal policy, which can fulfill any delivery ratio requirement vectors in the feasibility region.

Let $D_i^\eta(k)$ be the indicator random variable that is equal to 1 if client $i$ receives the packet from the base station during the interval $[kT, (k+1)T)$ by following policy $\eta$ and 0, otherwise. Recall our definition of feasibility region in Definition 2.1.3: a vector of delivery ratio requirements, $(q_i)_{i=1}^N$, belongs to the feasibility region if and only if there exists a policy, say $\eta$, in $\Pi$ such that $\hat{q}_i^\eta \geq q_i$ for all $i$ with probability 1 where $\hat{q}_i^\eta := \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} E[D_i^\eta(k)|\mathcal{H}_{k-1}^\eta]$.

Let $\hat{D}_i(k)) := E[D_i(k)|\mathcal{H}_{k-1}]$. Then, $d_i(k) := \sum_{j=0}^{k-1}(q_i - \hat{D}_i(j))$ denotes the delivery debt for each client $i$ at the beginning of frame $k$ and $Debt(k) = (d_i(k)^+)_{i=1}^N$ denotes the vector consisting of all the delivery debts at the beginning of frame $k$. The following policy is a max-weight policy, which maximizes $\sum_{i=1}^N d_i(k)^+ \cdot \hat{D}_i(k)$ during frame $k$:

---

FRAME-BASED MAX-WEIGHT POLICY:

(i) At the beginning of frame $k$, calculate the delivery ratio debt vector $Debt(k) = (d_i(k)^+)_{i=1}^N$.

(ii) Maximize $\sum_i d_i(k)^+ \cdot \hat{D}_i(k)$.

---

We prove that a max-weight policy is a feasibility optimal policy:

**Theorem 2.3.1** The frame-based max-weight policy proposed above is a feasibility optimal policy if all of the vectors on the outer boundary of the feasibility region are fulfilled by a stationary randomized policy, which does not depend on the packet delivery history.

*Proof.* See Appendix C □

According to Theorem 2.2.1, the frame-based greedy policy fulfills all of the vectors on the outer boundary of the feasibility region and is a stationary randomized policy. Hence, a max-weight policy is a feasibility optimal policy.

Maximizing $\sum_{i=1}^N d_i(k)^+ \cdot \hat{D}_i(k)$ is equivalent to maximizing the expected weighted sum throughput over the frame, where the weight vector is $Debt(k) = (d_i(k)^+)_{i=1}^N$ for that frame. From section 2.2, we know that the greedy policy with $\vec{\alpha} = (d_i(k)^+)_{i=1}^N$ achieves the maximum expected weighted sum throughput, EWST($\vec{\alpha} = (d_i(k)^+)_{i=1}^N$). Thus, the following policy is feasibility optimal.

---

FRAME-BASED GREEDY POLICY:

(i) At the beginning of frame $k$, calculate the expected delivery ratio debt vector $Debt(k) = [d_i(k)^+]$

(ii) At time slot $t$ serve client $j(t)$ such that $j(t) := \arg\max_{i\in\mathcal{N}(t)} d_i(k)^+ \cdot p_i$ where $\mathcal{N}(t)$ is the set of clients who have not received packets yet at time slot $t$.

In order to compare different feasibility optimal policies, we define the difference between the required delivery ratio and the achieved delivery ratio as a deadline miss ratio (DMR) function [2].

**Definition 2.3.1** Let $\tilde{q}_n(t)$ denote the delivery ratio achieved by client $n$ until time slot $t$. Then, we define the DMR function at time slot $t$ as:

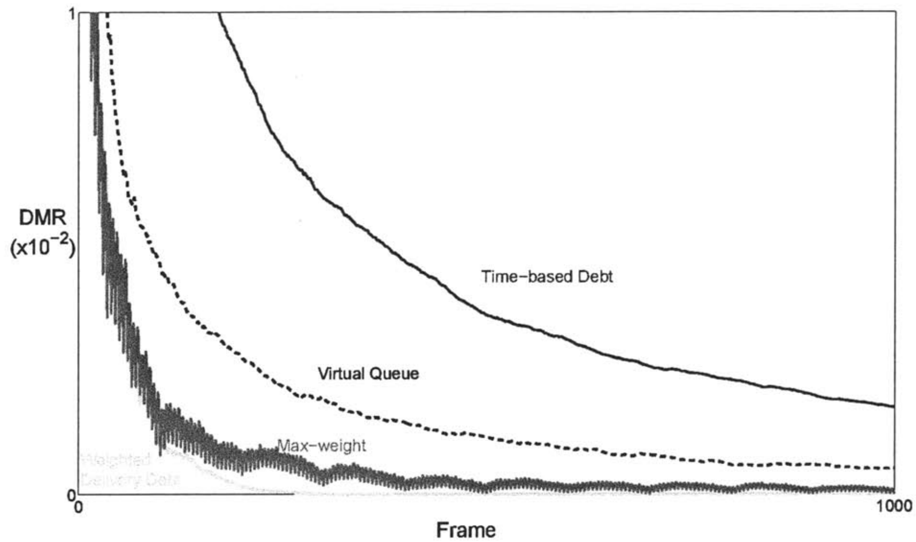
$$\mathrm{DMR}(t) := \frac{1}{N}\sum_{n=1}^{N}[q_n - \tilde{q}_n(t)]^+ .$$

In [2], Hou and Kumar propose two more feasibility optimal policies, called the *largest time-based debt first policy* and the *largest weighted-delivery debt first policy.* The largest time-based debt first policy is based on Lemma 2.2.2. Let $u_i(t)$ be an indicator R.V. that is equal to 1 if client $i$ is served during time slot $t$ and 0, otherwise. Then, the time-based debt at the beginning of frame $k$ is defined as $d_i^{\mathrm{time}}(k) := \sum_{t=0}^{kT-1}\left(\frac{q_i}{Tp_i} - u_i(t)\right)$. The largest time-based debt first policy assigns priorities in the decreasing order of $d_i^{\mathrm{time}}(k)$ and serve the clients according to the priority. The largest weighted-delivery debt first policy assigns priorities in the decreasing order of $d_i(k)/p_i$ instead of $d_i(k)^+ \cdot p_i$. We can also think of a policy which assigns priorities in the decreasing order of $d_i(k)^+$, denoted by "Virtual Queue Policy" in Figure 2-4. Table 2.1 summarizes the feasibility optimal policies.

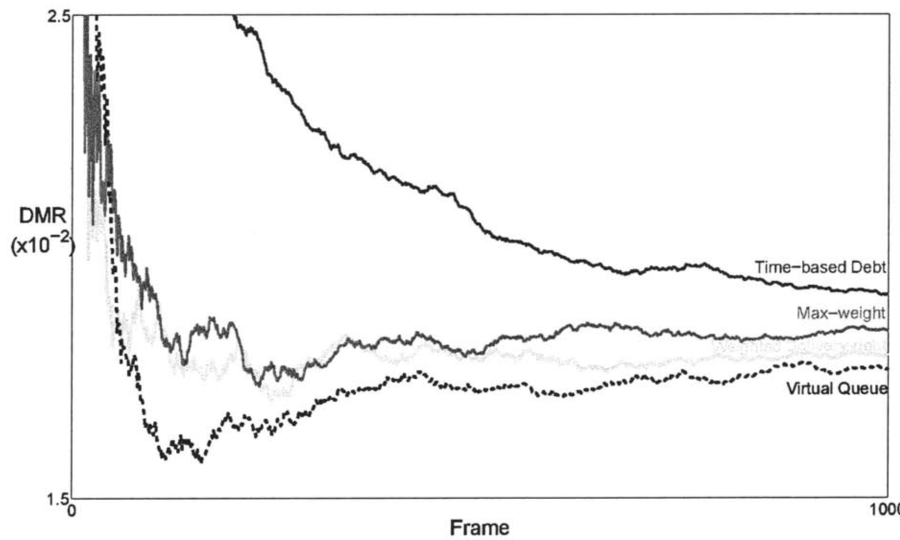
Table 2.1: Feasibility Optimal Policies

| Policy | Priority |
|---|---|
| Largest Time-based Debt First Policy | $d_i^{\mathrm{time}}(k)$ |
| Largest Weighted-delivery Debt First Policy | $d_i(k)/p_i$ |
| Max-weight Policy | $d_i(k)^+ \cdot p_i$ |
| Virtual Queue Policy | $d_i(k)^+$ |

As in [2], we consider two groups of clients, group $A$ and group $B$. Clients in group $A$ require a 99% delivery ratio, while clients in group $B$ require a 80% delivery ratio. The channel reliability of the $n$th client in both groups is assumed to be $(60+n)\%$. Using algorithm 1 of [2], when $T = 32$, we can show that a set of 11 group $A$ clients and 12 group $B$ clients is feasible but a set of 12 group $A$ clients and 12 group $B$ clients is not. For the feasible set consisting of 11 group $A$ clients and 12 group $B$ clients, $\lim_{t\to\infty}\mathrm{DMR}(t) = 0$ as shown in Figure 2-4a. Figure 2-4b shows how the




(a) Feasible Set




(b) Infeasible Set

Figure 2-4: Performance Comparison for Feasibility Optimal Policies for Instant Feedback

DMR function changes for the infeasible set consisting of 12 group $A$ clients and 12 group $B$ clients. Unlike the feasible set case, $\lim_{t\to\infty} \mathrm{DMR}(t) \neq 0$.

## 2.4 Summary

In this chapter, we presented the basic system model we use throughout the thesis. We characterized the feasibility region for a unicast network system with instant feedback by numerically maximizing the expected weighted sum throughput. We proved that the greedy policy, which maximizes the immediate expected reward, maximizes the expected weighted sum throughput. The greedy policy turned out to be feasibility optimal as well. We compared the performance of the greedy policy with that of the other feasibility optimal policies presented in [2]. The greedy policy performs nearly as well as the largest weighted-delivery debt first policy, which had the best performance.

Chapter 3

# Unicast with Delayed Feedback

One of the approaches to reliable data transmission over unreliable wireless channels is the use of Automatic Repeat reQuest (ARQ). When a client receives a packet successfully, it sends an ACK signal. If the transmission is not successful, the client sends a NACK signal. In some of the related works, Hou et al have focused on the network control problem of serving packets with hard deadlines in the unicast system with instant feedback [2, 4, 5].

Depending on the size and the structure of the network, however, the instant feedback may not be available or difficult to obtain. For example, in a typical satellite communications system, the delay can be more than 100ms. Even in a regular cell phone network, depending on the distance between a client and the base station, the feedback delay can be significant. In this chapter, we strengthen the model proposed in [2] to incorporate delayed feedback. We analyze how delayed feedback changes the feasibility region and develop a feasibility optimal policy under delayed feedback by formulating and solving a dynamic program.

The remainder of the chapter is organized as follows. We introduce the basic network model in Section 3.1. We characterize the feasibility region by solving the expected weighted sum throughput maximization in Section 3.2. We present feasibility optimal policies in Section 3.3. Simpler suboptimal policies are suggested and studied in Section 3.4.

## 3.1 System Model

Consider a wireless system where there is one base station sending data packets with delay constraints to $N$ wireless clients. This model is the same as the model used in Section 2.1, but extended to allow for delayed feedback. A time slot is long enough for a transmission of a packet to occur.

Thus, if a packet transmission is scheduled during a time slot and is successful, then the packet is delivered by the end of the time slot. Feedback, however, is delayed by a certain number of time slots. For example, if a packet is successfully delivered in time slot $t$ and the delay is $d$, then the base station will receive an ACK/NACK signal at the beginning of time slot $t+d+1$. For simplicity, we assume that delay $d$ does not depend on the client or on the frame. Thus, an ACK/NACK signal will always be received after $d$ time slots.

## 3.2 Feasibility Region

We use the numerical method developed in Chapter 2 to characterize the feasibility region in this section when the feedback signal is delayed. For a scheduling policy $\eta \in \Pi$, let $D_i^\eta(k)$ be a random variable that is equal to 1 if a packet is successfully transmitted to client $i$ during frame $k$ by following policy $\eta$ and 0, otherwise. To find a policy that maximizes the long-term expected weighted sum throughput, it is sufficient to consider a policy over only one frame because the packets arrive periodically at the beginning of each frame and any undelivered packets at the end of the frame are dropped. Hence, from now on, we only consider the first frame. For a given weight vector $(\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, the expected weighted sum throughput of policy $\eta$ during the first frame is $\sum_{i=1}^N E[\alpha_i D_i^\eta(0)]$. For each weight vector $\vec{\alpha}$, we find $\text{EWST}(\vec{\alpha}) := \max_{\eta\in\Pi} \sum_{i=1}^N E[\alpha_i D_i^\eta(0)]$. As before, by varying the weight vector $(\alpha_i)$ and solving the maximization problem, we can characterize the boundary of the feasibility region numerically.

In order to solve the maximization problem for the delayed feedback case, we use a Markov Decision Process. The initial state is $s_0 := (\{1, 2, \cdots, N\}, t = 0)$. Let $d$ denote the delay in feedback. Then, after the initial state, all the other states are characterized by three parameters: $(\mathcal{N}, u_{t-d} \cdots u_{t-1}, t)$ where $\mathcal{N}$ is the set of unserved clients at time slot $t$, $u_{t-d} \cdots u_{t-1}$ is the sequence of clients who have been served from $t-d$ to $t-1$, and $t$ is the current time slot. Note that $u_{t-d} \cdots u_{t-1}$ represents the clients who have been served but whose feedback have not arrived yet at the beginning of time slot $t$.

At the beginning, there are $N^{d+1}$ possible combinations of clients to serve from time slot 0 to time slot $d$ before time slot $d+1$ when the base station receives the first feedback from the transmission at time slot 0. Thus, if $A_{s_0}$ denotes the set of available actions from state $s_0$, then $A_{s_0} = \{u_0 u_1 \cdots u_d; u_i \in \{1, 2, \cdots, N\} \text{ for } i \in \{0, 1, \cdots, d\}\}$ and $|A_{s_0}| = N^{d+1}$.

Figure 3-1 shows the result of serving $u_0 u_1 \cdots u_d$. At the beginning of time slot $d+1$, the

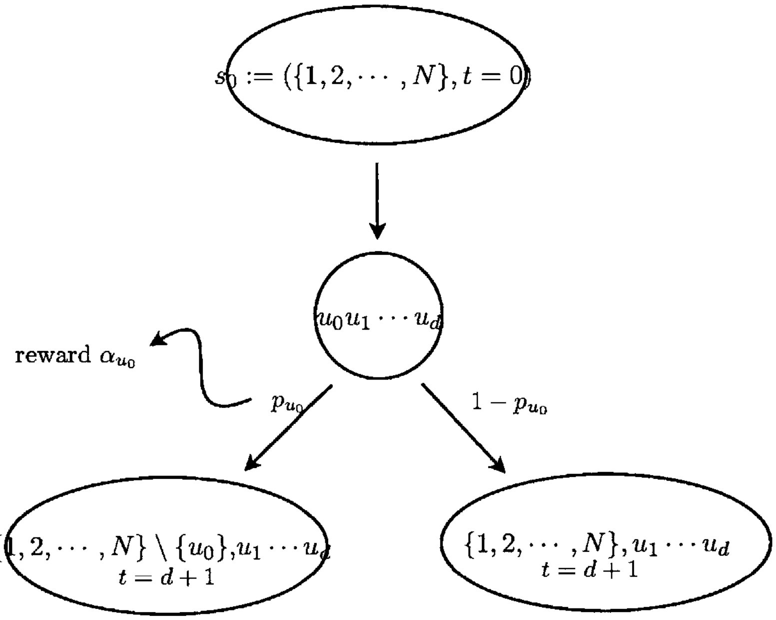



Figure 3-1: Markov Decision Process at Time $t = 0$ for Delayed Feedback;

base station receives the feedback regarding the transmission at time slot 0 since the feedback is delayed by $d$. With probability $p_{u_0}$, the transmission at time slot 0 is successful. Therefore, the state becomes $(\{1, 2, \cdots, N\} \setminus \{u_0\}, u_1 \cdots u_d, t = d + 1)$ and reward $\alpha_{u_0}$ is acquired. Similarly, with probability $1 - p_{u_0}$, the transmission at time slot 0 is not successful, so the state becomes $(\{1, 2, \cdots, N\}, u_1 \cdots u_d, t = d + 1)$ and reward 0 is acquired.

Example 3.2.1 applies Figure 3-1 to a unicast system in which there are two clients and the feedback is delayed by one time slot.

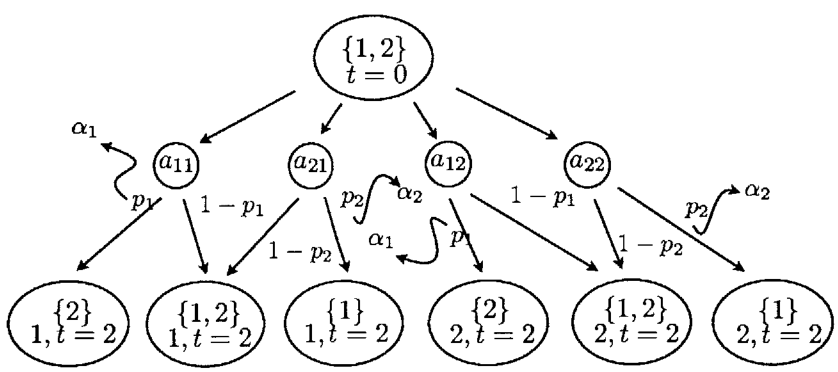



Figure 3-2: Markov Decision Process at Time $t = 0$ for Delayed Feedback; For simplicity, two clients and feedback delay of 1 time slot are assumed.

**Example 3.2.1** (2 Clients, $d = 1$) Figure 3-2 shows 4 actions which the base station can take at time slot 0 and time slot 1 when there are 2 clients and the delay is 1. $a_{ij}$ represents serving client $i$ in time slot 0 and client $j$ in time slot 1. Assume that $a_{11}$ is taken. Then, at the beginning of time slot $t = 2$, the base station receives feedback regarding the transmission during time slot 0. Hence, with probability $p_1$, the transmission is successful and the process reaches state $(\{2\}, 1, t = 2)$ with reward $\alpha_1 \geq 0$. Similarly, with probability $1 - p_1$, the transmission is not successful and the process reaches state $(\{1, 2\}, 1, t = 2)$.

Figure 3-3 shows the Markov Decision Process at time slot $d + 1 \leq t \leq T - 1$.

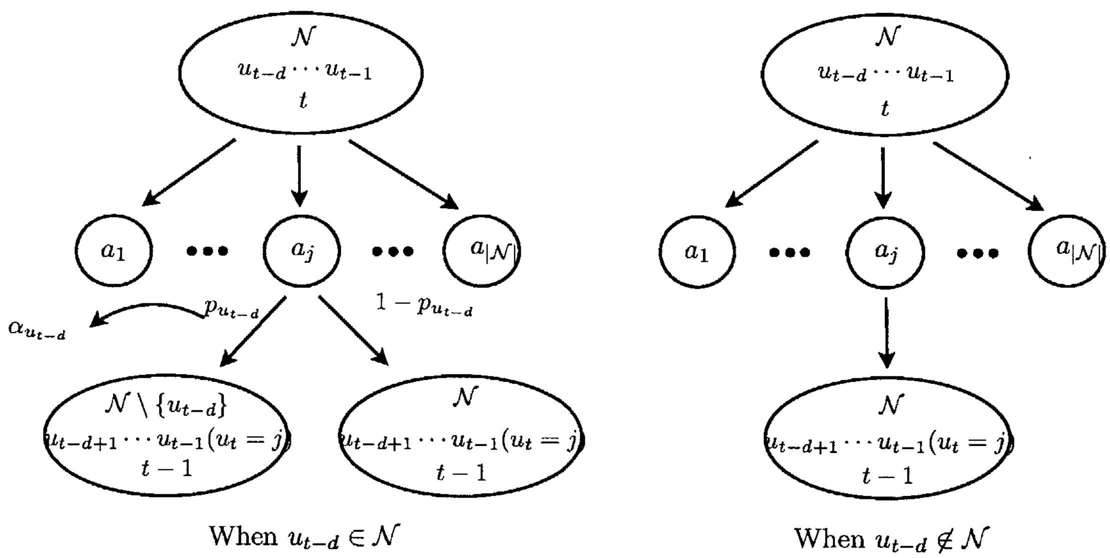



Figure 3-3: Markov Decision Process at Time $d + 1 \leq t \leq T - 1$ for Feedback Delayed by $d$ Time Slots;

Since the feedback is delayed by $d$ time slots, at the beginning of time slot $t$, the base station knows whether the transmission at time slot $t - d$ to client $u_{t-d}$ was successful or not. Let $a_j$ denote the action of serving client $j \in \mathcal{N}$. If $u_{t-d} \in \mathcal{N}$ and an ACK signal from time slot $t - d$ is received, then reward $\alpha_{u_{t-d}}$ is obtained since the transmission was successful and client $u_{t-d}$ has not received any packets beforehand. However, when a NACK signal is received, the transmission was not successful and no reward is acquired. When $u_{t-d} \notin \mathcal{N}$, regardless of the feedback signal (ACK/NACK), there is no reward because any duplicate packets are of no use to the client and thereby are dropped.

Similar to [1], we define a policy as a sequence of functions $\pi = \{\mu_0, \mu_{d+1}, \cdots, \mu_{T-1}\}$. For $t \geq d+1$, $\mu_t$ maps states $(\mathcal{N}, u_{t-d}, \cdots, u_{t-1}, t)$ for any $\mathcal{N} \subset \{1, \cdots, N\}$ and $u_{t-d}, ..., u_{t-1} \in \{1, \cdots, N\}$ into a client to serve at time slot $t$, i.e. $i(t) = \mu_t(\mathcal{N}, u_{t-d}, \cdots, u_{t-1}, t) \in \mathcal{N}$. When $t = 0$, $\mu_0$ maps states $(\mathcal{N}, 0)$ into a sequence of clients of length $d+1$, i.e. $(i(0), i(1), \cdots, i(d)) = \mu_0(\mathcal{N}, 0)$ where $i(t) \in \mathcal{N}$. The sequence $(i(0), i(1), \cdots, i(d))$ of clients represents the clients to be served until the first feedback is received.

Let $J_\pi(s_0)$ be the expected reward of policy $\pi$ starting at $s_0 := (\{1, \cdots, N\}, 0)$. Then, because of our reward definition, $J_\pi(s_0)$ is the expected weighted sum throughput of policy $\pi$. We want to find $\pi^*$, an optimal policy, which maximizes the total expected reward, i.e. the expected weighted sum throughput.

According to Proposition 1.3.1 in [1], the optimal cost $J^*(s_0)$ is equal to $J_0(s_0)$, given by the last step of the following algorithm (3.1)-(3.3), which proceeds backward in time from $T-1$ to 0:

$$J_T(\mathcal{N}, u_{T-d} \cdots u_{T-1}, T) = \sum_{i \in \mathcal{N}} \alpha_i \left[1 - (1-p_i)^{n_i}\right], \tag{3.1}$$

where $n_i$ is the number of time slots client $i$ has been scheduled from time slot $T-d$ to $T-1$. When client $i$ has not received packets yet during the frame, $i \in \mathcal{N}$ by definition. When client $i$ is served for $n_i$ time slots, the probability of at least one successful transmission is $1-(1-p_i)^{n_i}$. Therefore, the expected increase in the expected weighted sum throughput is $\alpha_i(1-(1-p_i)^{n_i})$.

For $d+1 \leq t \leq T-1$,

$$J_t(\mathcal{N}, u_{t-d} \cdots u_{t-1}, t) = \begin{cases} \alpha_{u_{t-d}} p_{u_{t-d}} + \max_{i \in \mathcal{N}} & \\ \left[p_{u_{t-d}} J_{t+1}(\mathcal{N} \setminus \{u_{t-d}\}, u_{t-d+1} \cdots u_{t-1}(u_t = i), t+1)\right. & \\ \left.+(1-p_{u_{t-d}}) J_{t+1}(\mathcal{N}, u_{t-d+1} \cdots u_{t-1}(u_t = i), t+1)\right] & \text{if } u_{t-d} \in \mathcal{N}; \\ \max_{i \in \mathcal{N}} \left[J_{t+1}(\mathcal{N}, u_{t-d+1} \cdots u_{t-1}(u_t = i), t+1)\right]; & \text{if } u_{t-d} \notin \mathcal{N} \neq \emptyset; \\ 0 & \text{if } \mathcal{N} = \emptyset. \end{cases} \tag{3.2}$$

and

$$\begin{aligned} J_0(\mathcal{N}, 0) = \max_{u_0, u_1, \cdots, u_d \in \mathcal{N}} & \left[\alpha_{u_0} p_{u_0} + p_{u_0} J_{d+1}(\mathcal{N} \setminus \{u_0\}, u_1, \cdots, u_d, d+1) + \right. \\ & \left. (1-p_{u_0}) J_{d+1}(\mathcal{N}, u_1, \cdots, u_d, d+1)\right] \end{aligned} \tag{3.3}$$

Furthermore, if $\mu_0^*(\mathcal{N}, 0)$ and $\mu_t^*(\mathcal{N}, u_{t-d}, \cdots, u_{t-1}, t)$ for $t \geq d+1$ are the arguments which solve the above dynamic program. Then, the policy $\pi^* = \{\mu_0^*, \mu_{d+1}^*, \cdots, \mu_{T-1}^*\}$ is optimal.

Because of the way we defined the rewards, $\pi^*$ contains the set of actions that maximizes the expected weighted sum throughput, and $J_{\pi^*}(\{1, \cdots, N\}, 0)$ is equal to $\text{EWST}(\vec{\alpha})$ for a given nonnegative weight vector $\vec{\alpha}$.

**Example 3.2.2** Consider the case in which there are two clients, the feedback is delayed by one time slot ($d = 1$), and there are two time slots per frame ($T = 2$). Since the feedback is delayed by one time slot, it is equivalent to no feedback case and there are four possible policies, namely $\pi_1 = \{11\}$, $\pi_2 = \{12\}$, $\pi_3 = \{21\}$, and $\pi_4 = \{22\}$.

$$\begin{aligned}
J_{\pi_1}(\{1,2\},0) &= \alpha_1(1-(1-p_1)^2) \\
J_{\pi_2}(\{1,2\},0) &= J_{\pi_3}(\{1,2\},0) = \alpha_1 p_1 + \alpha_2 p_2 \\
J_{\pi_4}(\{1,2\},0) &= \alpha_2(1-(1-p_2)^2)
\end{aligned}$$

Therefore, $J^*(\{1,2\},0) = \max[\alpha_1(1-(1-p_1)^2), \alpha_1 p_1 + \alpha_2 p_2, \alpha_2(1-(1-p_2)^2)]$.

We can interpret the expected weighted sum throughput as reward $\alpha_i$ is awarded whenever the packet transmission to client $i$ is successful. Then, we can think of a policy that consists of a sequence of functions that maximize the expected immediate reward.

**Definition 3.2.1** The greedy policy is a policy that consists of a sequence of functions that maximize the expected immediate reward.

The next example shows that the greedy policy is not always optimal when the feedback signal is delayed.

**Example 3.2.3** Consider the case in which there are two clients, the feedback is delayed by one time slot ($d = 1$), and there are 3 time slots per frame ($T = 3$). There are four possible actions to take at $t = 0$, namely $\mu_0^{(1)} = 11$, $\mu_0^{(2)} = 12$, $\mu_0^{(3)} = 21$, and $\mu_0^{(4)} = 22$. Let $\pi_i = \{\mu_0^{(i)}, \mu_2^*\}$ be a policy which takes $\mu_0^{(i)}$ in the first two time slots and the optimal action in the third time slot. Then,

$$\begin{aligned}
J_{\pi_1}(\{1,2\},0) &= \alpha_1 p_1 + p_1 J_2^*(\{2\},1,t=2) + (1-p_1)J_2^*(\{1,2\},1,t=2) \\
J_{\pi_2}(\{1,2\},0) &= \alpha_1 p_1 + p_1 J_2^*(\{2\},2,t=2) + (1-p_1)J_2^*(\{1,2\},2,t=2) \\
J_{\pi_3}(\{1,2\},0) &= \alpha_2 p_2 + p_2 J_2^*(\{1\},1,t=2) + (1-p_2)J_2^*(\{1,2\},1,t=2) \\
J_{\pi_4}(\{1,2\},0) &= \alpha_2 p_2 + p_2 J_2^*(\{1\},2,t=2) + (1-p_2)J_2^*(\{1,2\},2,t=2)
\end{aligned}$$

When $|\mathcal{N}| = 1$, there is only one client to serve and we obtain the following equations:

$$J_2^*(\{1\}, 2, t=2) = J_3^*(\{1\}, 1, t=3) = \alpha_1 p_1 \tag{3.4}$$

$$J_2^*(\{2\}, 1, t=2) = J_3^*(\{2\}, 2, t=3) = \alpha_2 p_2 \tag{3.5}$$

$$J_2^*(\{1\}, 1, t=2) = \alpha_1 p_1 + p_1 \cdot 0 + (1-p_1) J_3^*(\{1\}, 1, t=3) = \alpha_1 p_1 + (1-p_1)\alpha_1 p_1$$

$$J_2^*(\{2\}, 2, t=2) = \alpha_2 p_2 + p_2 \cdot 0 + (1-p_2) J_3^*(\{2\}, 2, t=3) = \alpha_2 p_2 + (1-p_2)\alpha_2 p_2$$

In (3.4), no reward is received during time slot 2 because client 2 has already received a packet. Similarly, in (3.5), no reward is received during time slot 2 because client 1 has already received a packet.

From state $(\{1,2\}, 1, t=2)$, there are two possible actions, namely serving client 1 ($\mu_2 = 1$) and serving client 2 ($\mu_2 = 2$):

$$J_{2,\mu_2=1}(\{1,2\}, 1, t=2) = \alpha_1 p_1 + (1-p_1)\alpha_1 p_1$$

$$J_{2,\mu_2=2}(\{1,2\}, 1, t=2) = \alpha_1 p_1 + \alpha_2 p_2.$$

Therefore, $J_2^*(\{1,2\}, 1, t=2) = \max[J_{2,\mu_2=1}(\{1,2\}, 1, t=2), J_{2,\mu_2=2}(\{1,2\}, 1, t=2)] = \alpha_1 p_1 + \max[(1-p_1)\alpha_1 p_1, \alpha_2 p_2]$. Similarly, $J_2^*(\{1,2\}, 2, t=2) = \alpha_2 p_2 + \max[(1-p_2)\alpha_2 p_2, \alpha_1 p_1]$.

When $\alpha_1 p_1 (1-p_1) \geq \alpha_2 p_2$, the greedy policy is equivalent to $\pi_1 = \{\mu_0^{(1)} = 11, \mu_2^*\}$. During the first time slot, the expected immediate reward of serving client 1 is $\alpha_1 p_1$ and that of serving client 2 is $\alpha_2 p_2$. Since $\alpha_1 p_1 \geq \alpha_1 p_1 (1-p_1)$, $\alpha_1 p_1 \geq \alpha_2 p_2$. Hence, the greedy policy serves client 1 during the first time slot. During the second time slot, the expected immediate reward of serving client 1 is $\alpha_1 p_1 (1-p_1)$ because the base station already served client 1 during the first time slot. Any duplicate packets are ignored by the client, so the reward for the transmission to client 1 is obtained during the second time slot only when the transmission during the first time slot failed. On the other hand, the expected immediate reward of serving client 2 during the second time slot is still $\alpha_2 p_2$. Therefore, when $\alpha_1 p_1 (1-p_1) \geq \alpha_2 p_2$, the greedy policy serves client 1 again during the second time slot. Assuming that $\alpha_1 p_1 (1-p_1) \geq \alpha_2 p_2$, we can show that $J_{\pi_1}$ and $J_{\pi_2}$ are bigger than or equal to $J_{\pi_3}$ and $J_{\pi_4}$. We calculate $J_{\pi_1} - J_{\pi_2}$ to decide which one is the biggest.

$$\begin{aligned} J_{\pi_1} - J_{\pi_2} &= -p_1 \alpha_2 p_2 (1-p_2) + (1-p_1)(\alpha_1 p_1 - \alpha_2 p_2 + (1-p_1)\alpha_1 p_1 - \alpha_1 p_1) \\ &= -p_1 \alpha_2 p_2 (1-p_2) + (1-p_1)(-\alpha_2 p_2 + (1-p_1)\alpha_1 p_1). \end{aligned} \tag{3.6}$$

From (3.6), we know that if $\alpha_1 p_1(1-p_1) - \alpha_2 p_2 < \frac{p_1}{1-p_1}\alpha_2 p_2(1-p_2)$, then $J_{\pi_2} > J_{\pi_1}$. Therefore, $\pi_2$ is the optimal policy when $0 \leq \alpha_1 p_1(1-p_1) - \alpha_2 p_2 \leq \frac{p_1}{1-p_1}\alpha_2 p_2(1-p_2)$.

As shown before, when $\alpha_1 p_1(1-p_1) \geq \alpha_2 p_2$, the greedy policy is equivalent to $\pi_1$. Hence, when $0 \leq \alpha_1 p_1(1-p_1) - \alpha_2 p_2 \leq \frac{p_1}{1-p_1}\alpha_2 p_2(1-p_2)$, the greedy policy is not optimal.

Note that the inequality $\alpha_1 p_1(1-p_1) - \alpha_2 p_2 \leq \frac{p_1}{1-p_1}\alpha_2 p_2(1-p_2)$ is likely to hold when $p_1$ is large. Then, it makes sense that when the inequality holds, $J_{\pi_2}$ is bigger than $J_{\pi_1}$. When $p_1$ is large, serving client 1 at time slot 0 and client 2 at time slot 1 does better than serving client 1 in a row since in the latter case it is likely that the second time slot is wasted.

**Lemma 3.2.1** As shown in Example 3.2.3, in the delayed feedback case, the greedy policy, which maximizes the expected immediate reward, is sub-optimal.

By solving (3.1)-(3.3) for a given weight vector ($\vec{\alpha}$), we can find the maximum expected weighted sum throughput, EWST($\vec{\alpha}$). By finding EWST($\vec{\alpha}$) for different weight vectors, we can numerically characterize the feasibility region as in Figure 3-4.

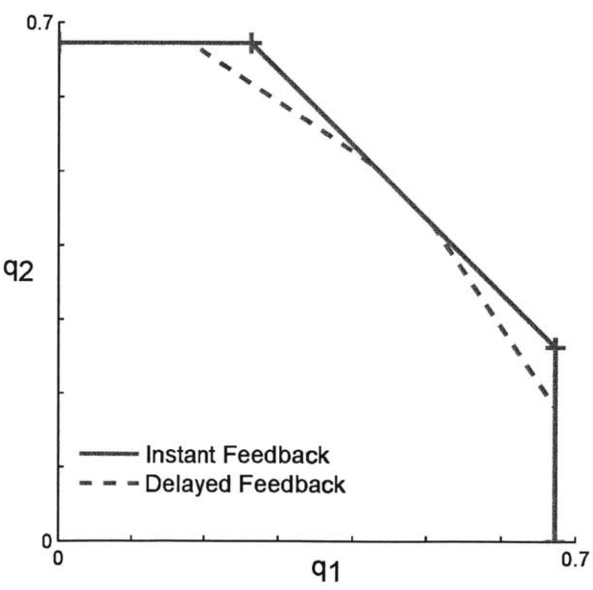



Figure 3-4: Delay yields smaller feasibility region; $p_1 = p_2 = 0.2$, $T = 5$ and the feedback is delayed by one time slot.

In order to determine the effect of delay on the feasibility region, we draw how EWST($\vec{\alpha}$) per client with weight vector $\vec{\alpha} = (1, 1, \cdots, 1)$ changes as feedback delay increases in Figure 3-5.

EWST$(1, \cdots, 1)$ under instant feedback is larger than that under delayed feedback. As delay gets larger, EWST$(1, \cdots, 1)$ under delayed feedback decreases and eventually become equal to

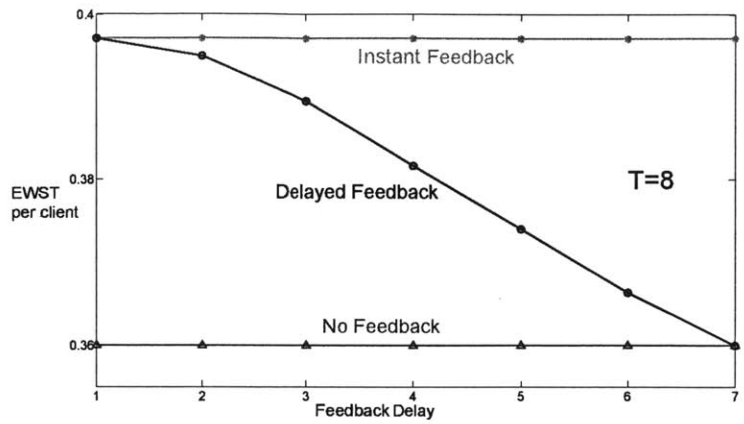



Figure 3-5: As the feedback delay increases, EWST($\vec{\alpha}$) per client decreases and eventually approaches the no feedback case; $\vec{\alpha} = (1, 1, \cdots, 1)$, $N = 4$, $T = 8$, and $p_i = 0.2$ for any client $i$

EWST$(1, \cdots, 1)$ under no feedback. As explained earlier, when $d \geq T - 1$, the feedback from the first time slot arrives after the end of the frame and thereby become useless.

Next, in Figure 3-6, we fix the delay and vary the number of clients.

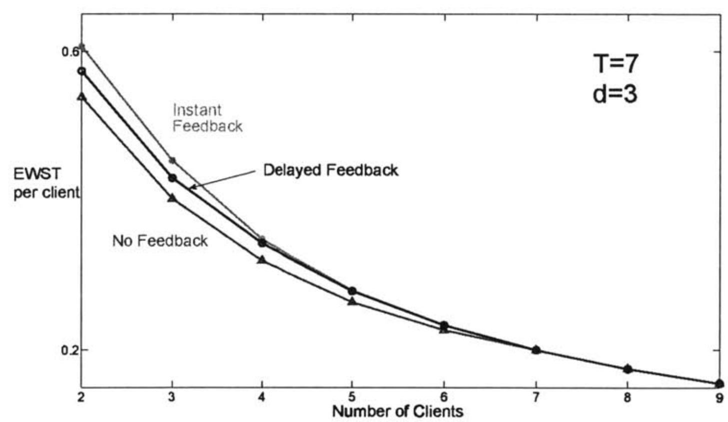



Figure 3-6: As the number of clients increases, EWST($\vec{\alpha}$) per client decreases for all three cases and eventually they all overlap; $\vec{\alpha} = (1, 1, \cdots, 1)$, $d = 3$, $T = 7$, and $p_i = 0.2$ for any client $i$

When the number of clients is small, EWST$(1, \cdots, 1)$ per client under instant feedback is larger than that under delayed Feedback, which in turn is larger than that under no feedback. However, as the number of clients increases, they all eventually overlap.

## 3.3 Feasibility Optimal Policy

After we characterize the feasibility region, we propose a feasibility optimal policy for the unicast system with delayed feedback so that we can serve any clients that are inside the feasibility region.

Let $D_i(k)$ be the indicator random variable that is equal to 1 if client $i$ receives the packet from the base station during the interval $[kT, (k+1)T)$ and 0, otherwise. Recall our definition of feasibility region in Definition 2.1.3: a vector of delivery ratio requirements, $(q_i)_{i=1}^N$, is fulfilled by policy $\eta \in \Pi$ if and only if $\hat{q}_i^\eta \geq q_i$ for all $i$ with probability 1 where $\hat{q}_i^\eta := \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} E[D_i^\eta(k)|\mathcal{H}_{k-1}^\eta]$.

Let $\hat{D}_i(k)) := E[D_i(k)|\mathcal{H}_{k-1}]$. Then, $d_i(k) := \sum_{j=0}^{k-1}(q_i - \hat{D}_i(j))$ denotes the delivery debt for each client $i$ at the beginning of frame $k$ and $Debt(k) = (d_i(k)^+)_{i=1}^N$ denotes the vector consisting of all the delivery debts at the beginning of frame $k$. The following policy is a max-weight policy, which maximizes $\sum_{i=1}^N d_i(k)^+ \cdot \hat{D}_i(k)$ during frame $k$:

---

FRAME-BASED MAX-WEIGHT POLICY:

(i) At the beginning of frame $k$, calculate the delivery ratio debt vector $Debt(k) = (d_i(k)^+)_{i=1}^N$.

(ii) Maximize $\sum_i d_i(k)^+ \cdot \hat{D}_i(k)$.

---

Using Theorem 2.3.1, we can show that a max-weight policy is a feasibility optimal policy. Note that maximizing $\sum_{i=1}^N d_i(k)^+ \cdot \hat{D}_i(k)$ is equivalent to maximizing the expected weighted sum throughput over the frame, where the weight vector is $Debt(k) = (d_i(k)^+)_{i=1}^N$ for that frame. From section 3.2, we know that solving the dynamic program (3.1)-(3.3) with $\vec{\alpha} = (d_i(k)^+)_{i=1}^N$ achieves the maximum expected weighted sum throughput, EWST$(\vec{\alpha} = (d_i(k)^+)_{i=1}^N)$. Thus, the following policy is feasibility optimal.

---

FRAME-BASED MAX- WEIGHT POLICY (UNICAST WITH DELAYED FEEDBACK):

(i) At the beginning of frame $k$, calculate the delivery ratio debt vector $Debt(k) = (d_i(k)^+)_{i=1}^N$.

(ii) Solve the dynamic program (3.1)-(3.3) with $\vec{\alpha} = (d_i(k)^+)_{i=1}^N$ as the weight vector.

---

In order to compare different feasibility optimal policies, we define the difference between the required delivery ratio and the achieved delivery ratio as a deadline miss ratio (DMR) function [2].

**Definition 3.3.1** Let $\tilde{q}_n(t)$ denote the delivery ratio achieved by client $n$ until time slot $t$. Then, we define the DMR function at time slot $t$ as:

$$\text{DMR}(t) := \frac{1}{N}\sum_{n=1}^{N}\left[q_n - \tilde{q}_n(t)\right]^+ .$$

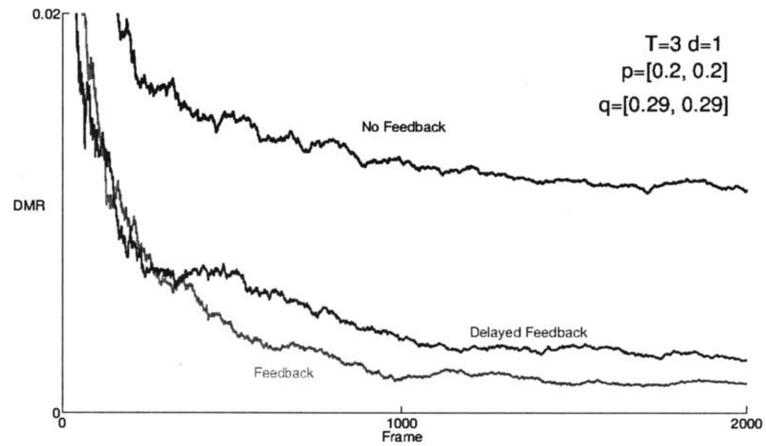



Figure 3-7: When there is a delay in feedback signal, DMR under delayed feedback decreases slower than the instant feedback case

Figure 3-7 shows how DMR changes when there is instant feedback, delayed feedback, or no feedback. The delivery ratio requirement vector $(0.29, 0.29)$ has been chosen to lie outside the feasibility region of the no feedback case but inside the feasibility regions of the instant feedback case and the delayed feedback case. The performance degrades as the feedback signal is delayed.

## 3.4 Sub-optimal Policies

The dynamic program (3.1)-(3.3) involves $\mathcal{O}(2^N N^d T)$ maximization, which may be too complicated to solve in real-time. Thus , in this section, we develop computationally efficient heuristic algorithms. A simple sub-optimal policy is the greedy algorithm defined in Definition 3.2.1. Figure 3-8 shows the feasibility region when the base station employs the sub-optimal greedy algorithm. While the feasibility region decreases, the reduction appears to be minimal.

**Definition 3.4.1** Frame-based Strict priority policy is a policy which assigns a priority to clients and the priority does not change within a frame. The base station starts serving a client with low priority only when it receives an ACK signal from a client with high priority.

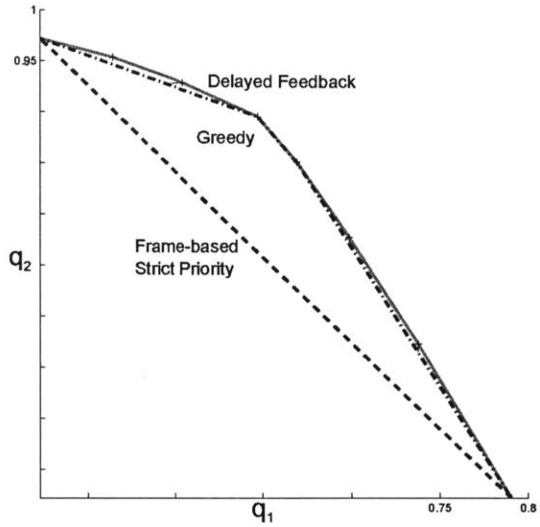



Figure 3-8: Feasibility Region of Unicast System with Delayed Feedback; $p_1 = 0.2$, $p_2 = 0.4$, $T = 7$ and the feedback is delayed by one time slot. The greedy algorithm indeed reduces the feasibility region but the decrease is not very significant

An example of the frame-based strict priority policies defined in Definition 3.4.1 is a policy which serves client 1 until the packet delivery to client 1 is successful. Figure 3-8 shows that the frame-based strict priority policies significantly reduce the feasibility region.

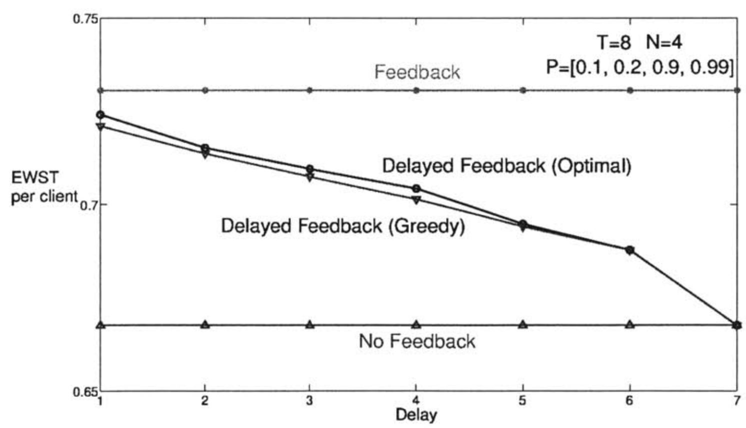



Figure 3-9: Greedy policy is suboptimal;

Figure 3-9 shows the performance of the greedy policy as the delay gets larger. The channel reliability vector has been chosen so that the greedy policy differs from the optimal policy. Note that as the delay gets bigger the difference between the optimal policy and the greedy policy decreases. It is because as the delay gets bigger the delayed feedback case becomes more similar to the no feedback case and the greedy policy is optimal in the no feedback case.

## 3.5 Summary

In this chapter, we studied a unicast network system with delayed feedback. We solved the maximization of the expected weighted sum throughput by using a dynamic program. Unlike the instant feedback case in Chapter 2, the greedy policy is not feasibility optimal. However, we show that the greedy policy does not decrease the feasibility region significantly and performs quite well compared to the other suboptimal policies.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

# Unicast with Time-varying Channels

In Chapter 2 and Chapter 3, we have considered static channels. However, in real world wireless networks, channel states vary over time due to node mobility, fading, or scattering.

Hou and Kumar modeled the time-varying wireless channels using an ergodic and irreducible Markov chain with a finite number of states in [4]. Assuming that the current channel state information is available, they characterize the feasibility region and a feasibility optimal policy by decouping the channel states.

Depending on the size and the structure of the network, however, the current channel state may not be available or difficult to obtain. For example, in some communications system, the current channel state is estimated from the past delivery results. Hence, in this chapter, we strengthen the model proposed in [4] to incorporate delayed channel state information. Instead of assuming that the current channel state information is available, we assume that the channel state information is delayed. We examine how the delay in the channel state information affects the feasibility region and develop feasibility optimal policies.

In addition to characterizing the feasibility region and feasibility optimal policies, Hou and Kumar studied utility maximization problem by considering the delivery ratio requirements as changeable parameters for static wireless networks [6]. In this chapter, we extend their approach and consider the utility maximization problem for time-varying wireless channels with delayed channel state information as well.

The remainder of the paper is organized as follows. We introduce the basic network model in Section 4.1. We characterize the feasibility region when the channel state information is delayed in Section 4.2. We prove that a max-weight policy is feasibility optimal in Section 4.3. We formulate and solve the utility maximization problem in Section 4.4.

## 4.1 System Model

Consider a wireless system where there is one base station sending data packets with delay constraints to $N$ wireless clients. This model is the same as the model used in Section 2.1, but extended to allow for time-varying channels.

Instead of assuming that channel reliability $p_i$ for client $i$ is fixed, we assume that the state of each channel changes according to a Markov chain such as the one shown in Figure 4-1, where channel transitions occur at the end of each frame. For simplicity, we only consider two-state Markov chains but the analysis can be extended easily to multi-state Markov chains. The channel state does not change within a frame.

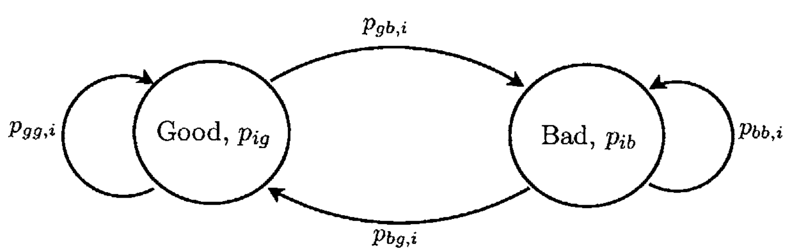



Figure 4-1: Time-varying Channel; we assume that each channel has two states: good and bad. $p_{gg,i}$ is the transition probability from good state to good state for client $i$ and the same labeling rule applies to the other transition probabilities.

We further assume that $p_{gb,i}, p_{bg,i} < 0.5$ for client $i$, so that each channel is positively correlated in time. We denote the $\tau$ step transition probabilities of the Markov chain in Figure 4-1 by $p^{(\tau)}_{gg,i}$, $p^{(\tau)}_{gb,i}$, $p^{(\tau)}_{bg,i}$, and $p^{(\tau)}_{bb,i}$. Then, as described in [7], the $\tau \geq 1$ step transition probabilities of the Markov chain in Figure 4-1 can be written as:

$$p^{(\tau)}_{gg,i} = \frac{p_{bg,i} + p_{gb,i}(1 - p_{bg,i} - p_{gb,i})^\tau}{p_{bg,i} + p_{gb,i}}, \tag{4.1}$$

$$p^{(\tau)}_{bb,i} = \frac{p_{gb,i} + p_{bg,i}(1 - p_{bg,i} - p_{gb,i})^\tau}{p_{bg,i} + p_{gb,i}}, \tag{4.2}$$

$$p^{(\tau)}_{gb,i} = \frac{p_{gb,i} - p_{gb,i}(1 - p_{bg,i} - p_{gb,i})^\tau}{p_{bg,i} + p_{gb,i}}, \tag{4.3}$$

$$p^{(\tau)}_{bg,i} = \frac{p_{bg,i} - p_{bg,i}(1 - p_{bg,i} - p_{gb,i})^\tau}{p_{bg,i} + p_{gb,i}}. \tag{4.4}$$

Let $p_{is}$ denote the channel reliability for client $i$ when the channel state is $s$. If we know that the channel state is "Good" $\tau$ frames ago, for example, then the expected channel reliability for client $i$ at the current frame, denoted by $\hat{p}_i^{(\tau)}$, can be written as:

$$\hat{p}_i^{(\tau)} = p_{gg,i}^{(\tau)} \cdot p_{ig} + p_{gb,i}^{(\tau)} \cdot p_{ib}. \tag{4.5}$$

## 4.2 Feasibility Region

For the ease of exposition, we first characterize the feasibility region when the current channel state information is available in Section 4.2.1 and extends the analysis to the feasibility region when the channel state information is delayed in Section 4.2.2.

### 4.2.1 Instant Channel State Information

Let $\Lambda_s$ denote the feasibility region under channel state $s$. After a sufficiently long time later, the probability of the channel state being $s$ is $\beta_s$, the steady state probability of channel state $s$. Then, we can show that the total feasibility region is $\sum_s \beta_s \Lambda_s$.

**Theorem 4.2.1** Let $S$ be the set of all the possible channel states and $\beta_s$ is the steady state probability of channel state $s \in S$. Then, the feasibility region can be expressed as $\Lambda := \sum_{s\in S} \beta_s \Lambda_s$.

*Proof.* Let $D_n(k)$ be an indicator variable that is equal to 1 if client $n$ receives a packet successfully during the $k$th frame, $[kT, (k+1)T)$ and 0, otherwise. Let $M_s(K)$ be the subset of frames with channel state $s$ during the first $K$ frames. Then, the empirical average throughput for client $n$ can be written as:

$$\overline{D_n(K)} := \frac{1}{K}\sum_{k=0}^{K-1} E[D_n(k)] = \sum_{s\in S} \frac{|M_s(K)|}{K}\left(\frac{1}{|M_s(K)|}\sum_{k\in M_s(K)} E[D_{ns}(k)]\right),$$

where $D_{ns}(k)$ be an indicator variable that is equal to 1 if the channel state is $s$ and $D_n(k) = 1$ and 0, otherwise.

First, we know that $(E[D_{ns}(k)])_{n=1}^N \in \Lambda_s$. Since $\Lambda_s$ is convex, the following holds:

$$\frac{1}{|M_s(K)|}\sum_{k\in M_s(K)} E[D_{ns}(k)] \in \Lambda_s.$$

Note that $\Lambda_s$ is closed by definition and so is $\sum_{s\in S} \beta_s \Lambda_s$ because it is a linear combination of closed sets. This implies that every limit point of the vector $\left(\overline{D_n(K)}\right)_{n=1}^N$ belongs to $\sum_{s\in S} \beta_s \Lambda_s$ because

$\lim_{K\to\infty} |M_s(K)|/K = \beta_s$. □

In Example 4.2.1, we apply Theorem 4.2.1 to characterize the two-client feasibility region when there are two channel states, i.e. $|S| = 2$.

**Example 4.2.1** Assume that under channel state $s$, the channel reliability for client 1 is $p_{1s}$ and the channel reliability for client 2 is $p_{2s}$. From Theorem 4 in [2], any point $(x, y)$ that satisfies the following equations is feasible.

$$\begin{aligned} \frac{x}{p_{1s}T} &\le 1 - I_{\{1\}} \\ \frac{y}{p_{2s}T} &\le 1 - I_{\{2\}} \\ \frac{x}{p_{1s}T} + \frac{y}{p_{2s}T} &\le 1 - I_{\{1,2\}}. \end{aligned}$$

where $I_L := \frac{E[\max\{0, T - \sum_{n\in L}\gamma_{ns}\}]}{T}$, $\gamma_{ns}$ is a geometrically distributed R.V. with parameter $p_{ns}$, and $L$ is a subset of clients.

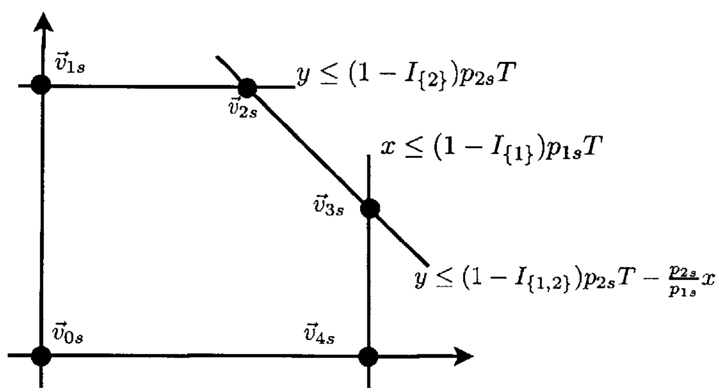



Figure 4-2: $\Lambda_s$ typically looks like this when there are only two clients.

As shown in Figure 4-2, $\Lambda_s$ can be expressed as a convex hull of five points, namely $\vec{v}_{0s} = (0, 0)$, $\vec{v}_{1s} = (0, (1 - I_{\{2\}})p_{2s}T)$, $\vec{v}_{2s} = ((I_{\{2\}} - I_{\{1,2\}})p_{1s}T, (1 - I_{\{2\}})p_{2s}T)$, $\vec{v}_{3s} = ((1 - I_{\{1\}})p_{1s}T, (I_{\{1\}} - I_{\{1,2\}})p_{2s}T)$, $\vec{v}_{4s} = ((1 - I_{\{1\}})p_{1s}T, 0)$. For any real number $\beta_s$, $\beta_s\Lambda_s = \beta_s\text{conv}\{\vec{v}_{0s}, \cdots, \vec{v}_{4s}\} = \text{conv}\{\beta_s\vec{v}_{0s}, \cdots, \beta_s\vec{v}_{4s}\}$. Here, conv$(A)$ denotes the convex hull of a set $A$.

According to Theorem 1.1.2 in [10], Minkowski addition and convex hull are commutative:

**Theorem 4.2.2** If $A, B \in \mathbb{R}^n$, then conv$(A + B)$=conv$(A)$+conv$(B)$.

Because $\beta_s \Lambda_s$ is a convex hull of points, using Theorem 4.2.2, we obtain:

$$\beta_{s_1}\Lambda_{s_1} + \beta_{s_2}\Lambda_{s_2} = \text{conv}(\{\beta_{s_1}\vec{v}_{0s_1}, \cdots, \beta_{s_1}\vec{v}_{4s_1}\} + \{\beta_{s_2}\vec{v}_{0s_2}, \cdots, \beta_{s_2}\vec{v}_{4s_2}\})$$

Figure 4-3 shows the result of the sum of the two convex hulls. Note that when the channel state does not change, each extreme point can be achieved by a strict priority policy [13]. For example, in Figure 4-2, point $\vec{v}_{2s}$ is achieved by a priority policy that serves client 2 first and then client 1 only when the transmission to client 2 is successful. Therefore, Theorem 4.2.1 and Theorem 4.2.2 imply that an extreme point of the feasibility region for time-varying channels is achieved by a channel-state-dependent strict priority policy. For instance, in Figure 4-3, giving strict priority to client 1 regardless of the channel states yields $\beta_{s_1}\vec{v}_{3s_1} + \beta_{s_2}\vec{v}_{3s_2}$, giving strict priority to client 2 regardless of the channel states yields $\beta_{s_1}\vec{v}_{2s_1} + \beta_{s_2}\vec{v}_{2s_2}$, and giving strict priority to client 1 when the channel state is $s_1$ and to client 2 when the channel state $s_2$ yields $\beta_{s_1}\vec{v}_{3s_1} + \beta_{s_2}\vec{v}_{2s_2}$.

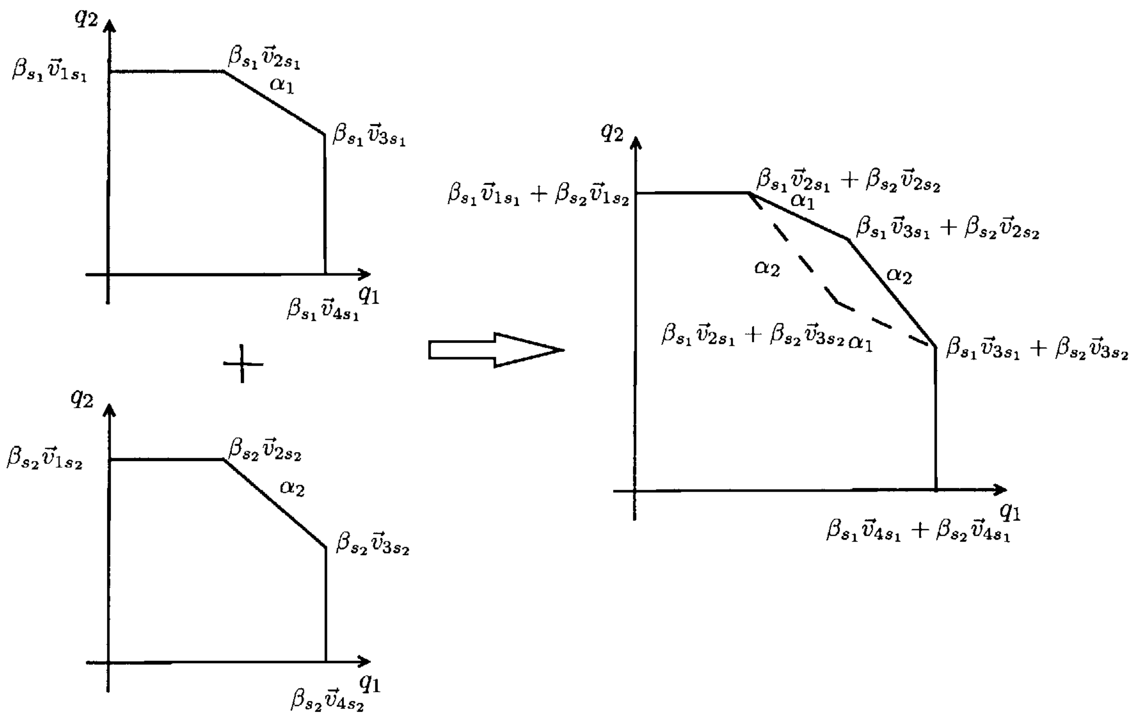



Figure 4-3: Sum of Two Feasibility Regions; The graphs on the left represent $\beta_{s_1}\Lambda_{s_1}$ and $\beta_{s_2}\Lambda_{s_2}$ where $\beta_{s_i}$ is a positive constant and $\Lambda_{s_i}$ is a feasibility region when the channel state is $(p_{1s_i}, p_{2s_i})$. $\alpha_i := \frac{p_{2s_i}}{p_{1s_i}}$ is the absolute value of the slope of the line segment from $\beta_{s_i}\vec{v}_{2s_i}$ to $\beta_{s_i}\vec{v}_{3s_i}$. As the graph on the right side shows, when $\alpha_1 \leq \alpha_2$, the resulting sum is the convex hull of $\{0, \beta_{s_1}\vec{v}_{1s_1} + \beta_{s_2}\vec{v}_{1s_2}, \beta_{s_1}\vec{v}_{2s_1} + \beta_{s_2}\vec{v}_{2s_2}, \beta_{s_1}\vec{v}_{3s_1} + \beta_{s_2}\vec{v}_{2s_2}, \beta_{s_1}\vec{v}_{3s_1} + \beta_{s_2}\vec{v}_{3s_2}, \beta_{s_1}\vec{v}_{4s_1} + \beta_{s_2}\vec{v}_{4s_2}\}$.

### 4.2.2 Delayed Channel State Information

Instead of knowing the current channel state, the base station may only know the channel state $\tau$ frames ago. In this section, we characterize the feasibility region when the channel state information is delayed. For simplicity, assume that the delay is constant, i.e. at the beginning of frame $k$, the base station knows the channel state of frame $k - \tau$.

Let $\Lambda_s^{(\tau)}$ denote the feasibility region of the case in which the base station knows that $\tau$ frames ago, the channel state is $s$. Then, the overall feasibility region can be written as $\Lambda^{(\tau)} = \sum_{sS} \beta_s \Lambda_s^{(\tau)}$.

**Theorem 4.2.3** Let $\mathcal{S}$ be the set of all possible channel states and $\beta_s$ be the steady state probability of channel state $s \in \mathcal{S}$. Assume that the base station knows only the channel state $\tau$ frames ago. Let $\Lambda_s^{(\tau)}$ denote the feasibility region of the case in which the base station knows that $\tau$ frames ago, the channel state is $s$. Then, the feasibility region when the channel state information is delayed by $\tau$ frames can be expressed as $\Lambda^{(\tau)} := \sum_{s \in \mathcal{S}} \beta_s \Lambda_s^{(\tau)}$.

*Proof.* The proof is essentially the same as the proof of Theorem 4.2.1. In the proof of Theorem 4.2.1, replace $\Lambda_s$ with $\Lambda_s^{(\tau)}$. □

Hence, we need to characterize $\Lambda_s^{(\tau)}$. Consider a situation in which the base station knows that channel state is $s$, $\tau$ frames ago. Let $u_{ns}$ be the number of time slots allocated to client $n$ in such a situation. Following the notations from [11], a performance measure $x_s^v(n)$ is defined to be equal to $E[u_{ns}]$ under policy $v$. Let $\mathbf{x}_s^v := (x_s^v(n))_{n=1}^N$ be the performance vector. For any given permutation $\pi$ of the $N$ elements of $\{1, \cdots, N\}$, we can view $\pi$ as an absolute priority policy that assigns priority to the clients according to the permutation $\pi$, i.e. $\pi(1)$ is served first,...,$\pi(N)$ is served last. We want to show that the performance vector $\mathbf{x}_s$ satisfies strong conservation laws defined in [11].

**Definition 4.2.1** (Definition 5 in [11]) Consider a queueing system serving $N$ clients. Let $x_i$ be a performance measure of interest for client $i$. The vector $(x_1, \cdots, x_N)$ is said to satisfy the strong conservation laws if the following two conditions hold:

- The total performance of all clients, $\sum_{i=1}^N x_i$, is invariant under any admissible policy.
- The performance over each subset $F$ of clients, $\sum_{i \in F} x_i$, is optimized by any admissible policy that prioritizes clients in $F$ over those that are not in $F$.

**Lemma 4.2.1** Let $u_{ns}^{(\tau)}(k) \in \{0, 1, \cdots, T\}$ be the number of time slots allocated to client $n$ in the $k$th frame when the base station knows that channel state is $s$, $\tau$ frames ago. Define

$x_s(n) = E[u_{ns}^{(\tau)}(k)]$. Then, the performance vector $(x_s(n))_{n=1}^N$ satisfies the strong conservation laws defined in Definition 4.2.1.

*Proof.* Define $f_s(L) = \sum_{s' \in S} E[\min(T, \sum_{n \in L} r_{ns'})] \cdot p_{s \to s'}^{(\tau)}$ for $L \subset \{1, \cdots, N\}$ where $p_{s \to s'}^{(\tau)}$ is $\tau$ step transition probability from state $s$ to state $s'$. For any $L \subset \{1, \cdots, N\}$, define $x_s(L) := \sum_{i \in L} x_s(i)$. Consider $\pi$ such that $L = \{\pi(1), \cdots, \pi(|L|)\}$, i.e. a priority rule which prioritizes the clients in $L$ over the clients in $\{1, \cdots, N\} \setminus L$. Then, the following holds trivially:

$$f_s(L) = x_s^{\pi}(L) \tag{4.6}$$

For any non-idling and non-anticipative policy $v$, we can show that the following holds:

$$x_s^v(L) \leq f_s(L), \quad \forall L \subset \{1, \cdots, N\}; \quad x_s^v(\{1, \cdots, N\}) = f_s(\{1, \cdots, N\}). \tag{4.7}$$

Therefore, the performance vector $\mathbf{x}_s$ satisfies strong conservation laws: (4.6) and (4.7) □

From $f_s$, we can define a polyhedron $\mathcal{B}(f_s)$ in a following manner:

$$\mathcal{B}(f_s) := \{\mathbf{x} \geq 0 | x(L) \leq f_s(L), L \subset \{1, \cdots, N\}; x(\{1, \cdots, N\}) = f_s(\{1, \cdots, N\})\}. \tag{4.8}$$

Then from Theorem 1 of [11], we know that $\mathcal{B}(f_s)$ is the performance space and the vertices of $\mathcal{B}(f_s)$ are achieved by the strict priority rules.

**Lemma 4.2.2** (Theorem 1 in [11]) If a performance measure vector, $(x_1, \cdots, x_N)$, satisfies the strong conservation laws, its achievable region of the performance measure vectors is the base of a polymatroid, a polyhedron with the property that there is a one-to-one correspondence between its vertices and the set of strict priority policies.

Therefore, each vertices of $\Lambda_s^{(\tau)}$ are achieved by the strict priority rules as well. For instance, consider a two-client case, when $T > 1$ and $0 < p_{is} < 1$ for $i \in \{1, 2\}$, $\Lambda_s$ is a convex hull of five points and can be written as $\Lambda_s = \text{conv}\{\vec{v}_{0s}, \vec{v}_{1s}, \vec{v}_{2s}, \vec{v}_{3s}, \vec{v}_{4s}\}$. Then, $\Lambda_s^{(\tau)} = \text{conv}\{\sum_{s'} p_{s \to s'}^{(\tau)} \vec{v}_{0s}, \cdots, \sum_{s'} p_{s \to s'}^{(\tau)} \vec{v}_{4s}\}$.

Figure 4-4 shows an example of the feasibility regions with various delays. As expected, the longer the delay is, the smaller the feasibility region becomes. If the delay is sufficiently long, the feasibility region is the same as the case in which the base station does not know the channel state at all.

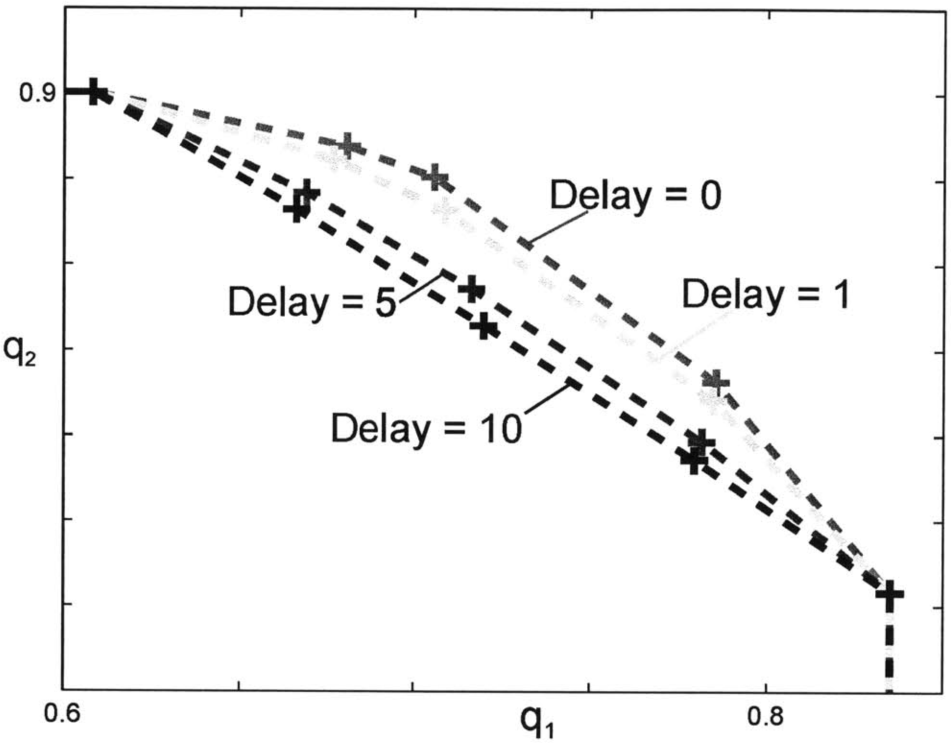



Figure 4-4: Feasibility Region with Delayed Channel State Information; T=5, the channel state of client 1 alternates between "G" with channel reliability 0.7 and "B" with channel reliability 0.2 ($p_{gb,1} = p_{bg,1} = 0.1$) and the channel state of client 2 alternates between "G" with channel reliability 0.5 and "B" with channel reliability 0.3 ($p_{gb,2} = p_{bg,2} = 0.1$).

## 4.3 Feasibility Optimal Policy

After we characterize the feasibility region, we propose a feasibility optimal policy for the unicast system with delayed feedback so that we can serve any clients that are inside the feasibility region.

Let $D_i(k)$ be the indicator random variable that is equal to 1 if client $i$ receives the packet from the base station during the interval $[kT, (k+1)T)$ and 0, otherwise. Recall our definition of feasibility region in Definition 2.1.3: given a vector of delivery ratio requirements, $(q_i)_{i=1}^N$, the system is fulfilled if and only if $\hat{q}_i \geq q_i$ for all $i$ with probability 1 where $\hat{q}_i := \liminf_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1} E[D_i(k)|\mathcal{H}_{k-1}]$ and $\mathcal{H}_{k-1}$ is the history of packet deliveries up to and including frame $k-1$.

Let $\hat{D}_i(k)) := E[D_i(k)|\mathcal{H}_{k-1}]$. Then, $d_i(k) := \sum_{j=0}^{k-1}(q_i - \hat{D}_i(j))$ denotes the delivery debt for each client $i$ at the beginning of frame $k$ and $Debt(k) = (d_i(k)^+)_{i=1}^N$ denotes the vector consisting of all the delivery debts at the beginning of frame $k$. The following policy is a max-weight policy $\eta^0$ that maximizes $\sum_{i=1}^N d_i(k)^+ \cdot \hat{D}_i(k)$ during frame $k$:

---

FRAME-BASED MAX-WEIGHT POLICY:

(i) At the beginning of frame $k$, calculate the delivery ratio debt vector $Debt(k) = (d_i(k)^+)_{i=1}^N$.

(ii) Maximize $\sum_i d_i(k)^+ \cdot \hat{D}_i(k)$.

---

From Theorem 2.3.1, we know that a max-weight policy is a feasibility optimal policy. Since the greedy policy achieves max-weight, the greedy policy is feasibility optimal as well. Therefore, the following algorithm achieves max-weight and thereby is feasibility optimal.

---

FRAME-BASED GREEDY POLICY:

(i) At the beginning of frame $k$, calculate the expected delivery ratio debt vector $D(k) = (d_i(k)^+)_{i=1}^N$

(ii) Serve the client with the highest $d_i(k)^+ \cdot \hat{p}_i(k)$.

---

Here, $\hat{p}_i(k)$ is the expected channel reliability of client $i$ during frame $k$. For example, if the channel state during frame $k-1$ is $s_1$, then $\hat{p}_i(k) = \sum_s p_{s_1 \to s}^{(1)} p_{is}$.

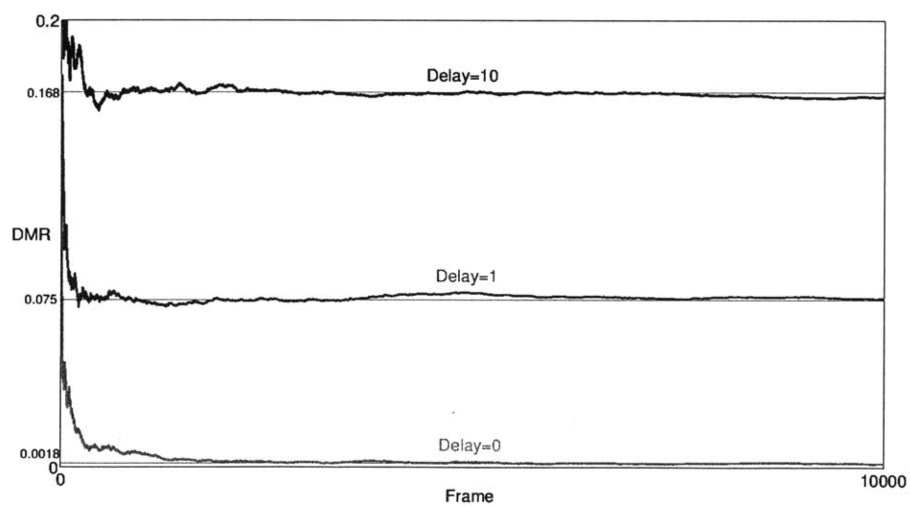



Figure 4-5: As the delay of the channel state information gets larger, the performance of the max-weight policy gets worse;

As in [2], we define the difference between the required delivery ratio and the achieved delivery ratio as a deadline miss ratio (DMR) function. Let $\tilde{q}_n(t)$ denote the delivery ratio achieved by client $n$ until time slot $t$. Then, the DMR function at time slot $t$ can be written as:

$$\mathrm{DMR}(t) = \frac{1}{N}\sum_{n=1}^{N}\left[q_n - \tilde{q}_n(t)\right]^+ .$$

We consider the scenario presented in [4] to simulate how the delay of the channel state information affects the performance of the max-weight policy in Figure 4-5. The channel reliability in the good state is 1 and the channel reliability in the bad state is 0.2. There are two groups of clients called group A and group B. Both groups consist of 19 clients each. Clients in group A require 0.9 delivery ratios and clients in group B require 0.6 delivery ratios. For the $n$th client in each group, $p_{gb,n} = \frac{0.5}{n+3}$ and $p_{bg,n} = \frac{0.5(n+2)}{n+3}$.

## 4.4 Utility Maximization

As in [6], considering the delivery ratio for each client as a changeable parameter, we can study the problem of maximizing the total utility of a wireless communication system. Let $g_n$ be the utility function for client $n$. We assume that $g_n$ is concave, continuous, and increasing in the domain of our interest, $[0, 1]$. Since $[0, 1]$ is compact, we can guarantee that $\exists M > 0$ such that $|g_n(x)| \leq M$ for $\forall x \in [0, 1]$. Here, the assumption that $g_n$ is concave reflects the law of diminishing returns. Given a feasibility region $\Lambda$, a typical total utility maximization problem can be written as:

$$\max \quad \sum_{i=1}^{N} g_i(\overline{d}_i) \tag{4.9}$$

$$\text{subject to} \quad (\overline{d}_i),\ (q_i)\ \in \Lambda \tag{4.10}$$

$$\overline{d}_i \geq q_i, \quad \forall i \in \{1, \cdots, N\}. \tag{4.11}$$

Here, we maximize the total utility of a system (4.9) while guaranteeing minimum throughput for each client (4.11). Following the approach outlined in Chapter 5 of [9], we introduce a vector of auxiliary variables $\vec{\gamma}(k) = (\gamma_1(k), \cdots, \gamma_N(k))$ and consider the following modified problem:

$$
\begin{aligned}
\max \quad & \overline{g(\vec{\gamma})} && (4.12)\\
\text{subject to} \quad & (\overline{d}_i),\ (q_i)\ \in \Lambda && (4.13)\\
& \overline{d}_i \geq q_i, \quad \forall i \in \{1, \cdots, N\} && (4.14)\\
& \overline{\gamma}_j \leq \overline{d}_j, \quad \forall j \in \{1, \cdots, N\}. && (4.15)
\end{aligned}
$$

Here, we define $g(\vec{\gamma}) := \sum_{i=1}^{N} g_i(\gamma_i)$. Following the drift-plus-penalty method outlined in Chapter 4 of [9], we enforce $\overline{d}_i \geq q_i$ and $\overline{\gamma}_i \leq \overline{d}_i$ in the transformed problems (4.14)-(4.15) with virtual queues $Z_i(k)$ and $G_j(k)$:

$$
\begin{aligned}
Z_i(k+1) &= \max[Z_i(k) + q_i - d_i(k), 0]\ ,\ \forall i \in \{1, \cdots, N\} && (4.16)\\
G_j(k+1) &= \max[G_j(k) + \gamma_j(k) - d_j(k), 0]\ ,\ \forall j \in \{1, \cdots, N\} && (4.17)
\end{aligned}
$$

Define $\mathbf{\Theta}(k) := [\mathbf{Z}(k), \mathbf{G}(k)]$ and define the Lyapunov function:

$$
L(\mathbf{\Theta}(k)) := \frac{1}{2}\left[\sum_{i=1}^{N} w_i (Z_i^2(k) + G_i^2(k))\right] \tag{4.18}
$$

From (5.21) in [9], we can show that:

$$
\begin{aligned}
\Delta(\mathbf{\Theta}(k)) - V \cdot E[g(\vec{\gamma}(k))|\mathbf{\Theta}(k)] \leq D - V \cdot E[g(\vec{\gamma}(k))|\mathbf{\Theta}(k)] + \sum_{i=1}^{N} w_i Z_i(k) \cdot E[q_i - d_i(k)|\mathbf{\Theta}(\mathbf{k})] \\
+ \sum_{j=1}^{N} w_j G_j(k) \cdot E[\gamma_j(k) - d_j(k)|\mathbf{\Theta}(\mathbf{k})] \qquad (4.19)
\end{aligned}
$$

where $D$ is a finite constant. To minimize the right hand side of (4.19), we perform the following:

(Auxiliary Variables) For each frame $k$, observe $\mathbf{G}(k)$ and choose $\vec{\gamma}(k)$ to solve the following convex problem:

$$
\begin{aligned}
\max \quad & V g(\vec{\gamma}(k)) - \sum_{i=1}^{N} w_i G_i(k) \gamma_i(k) && (4.20)\\
\text{subject to} \quad & 0 \leq \gamma_i(k) \leq 1 \quad \forall i \in \{1, \cdots, N\} && (4.21)
\end{aligned}
$$

(Decision) For each frame $k$, choose a policy so that it maximizes:

$$\sum_{i=1}^{N} w_i(Z_i(k) + G_i(k))E[d_i(k)|\Theta(k)] \tag{4.22}$$

Suppose that the base station knows that the channel state is $s$, $\tau$ frames ago. Then, at the current time slot, the expected channel reliability for client $n$ can be written as:

$$\hat{p}_{ns}^{(\tau)} = \sum_{t \in \mathcal{S}} p_{st}^{(\tau)} \cdot p_{nt} \tag{4.23}$$

where $\mathcal{S}$ is the set of all possible states and $p_{st}^{(\tau)}$ is $\tau$ step transition probability from state $s$ to state $t$ and $p_{nt}$ is the channel reliability for client $n$ under channel state $t$. Then, (4.22) can be maximized via a strict priority policy such that a client with higher $(Z_i(k) + G_i(k))w_i\hat{p}_{is}^{(\tau)}$ is served earlier.

Consider a frame where the base station knows that the channel state is $s$, $\tau$ frame ago. Let $u_{ns}$ be the number of time slots allocated to client $n$ in such a case. Then, we can show that a performance measure $x_s(n) := E[u_{ns}]$ satisfies strong conservation laws defined in [11]. Theorem 1 of [11] says that the performance region of $\mathbf{x}_s := (E[u_{ns}])_n$ is a polymatroid, denoted by $\Omega_s$. The expectation in (4.22) can be written as:

$$E[d_i(k)|\Theta(k)] = E[E[d_i(k)|\Theta(k), s(k-\tau) = s]] = E[\hat{p}_{is}^{(\tau)} E[u_{is}(k)|\Theta(k), s(k-\tau) = s]] \tag{4.24}$$

Therefore, at the beginning of frame $k$, after observing the channel state $\tau$ frames ago, denoted by $s(k-\tau) = s$ and the queues, denoted by $\Theta(k)$, maximizing (4.22) can be achieved by solving the following:

$$\max \quad \sum_{i=1}^{N} w_i \hat{p}_{is}^{(\tau)} (Z_i(k) + G_i(k)) E[u_{is}(k)] \tag{4.25}$$

$$\text{subject to} \quad E[u_{is}(k)] \in \Omega_s \tag{4.26}$$

Then, because of Theorem 3 in [13], the optimal solution to (4.25) is obtained by the permutation corresponding to the decreasing order of $(Z_i(k) + G_i(k))w_i\hat{p}_{is}^{(\tau)}$. Therefore, the total utility maximizing policy can be summarized as follows:

---

Utility-Optimal Policy (UOP)

(i) Fix some finite parameter $w_i > 0$. At the beginning of frame $k$, observe $\mathbf{G}(k)$ and choose $\vec{\gamma}(k)$ to solve the convex problem (4.20)-(4.21).

(ii) Serve clients in the decreasing order of $(Z_i(k) + G_i(k))w_i\hat{p}_{is}^{(\tau)}$ when the base station knows that the channel state $\tau$ frames ago is $s$. The base station serves the next client only when the transmission to the previous client is successful.

(iii) Update $Z_i(k)$ and $G_i(k)$ according to (4.16) and (4.17).

---

Consider a two-client wireless network with the channel state space $S = \{gg, bb\}$. When the channel state is in $gg$, the channel reliability for client 1 is $p_{1,gg}$=0.8 and that for client 2 is $p_{2,gg} = 0.5$. Similarly, when the channel state is in $bb$, the channel reliability vector is $(p_{1,bb}, p_{2,bb}) = (0.3, 0.2)$. The transition probability from $gg$ to $bb$ is $p_{gg,bb} = 0.25$ and that from $bb$ to $gg$ is $p_{bb,gg} = 0.25$ so that $\beta_{gg} = \beta_{bb} = 0.5$. Each frame consists of 5 time slots.

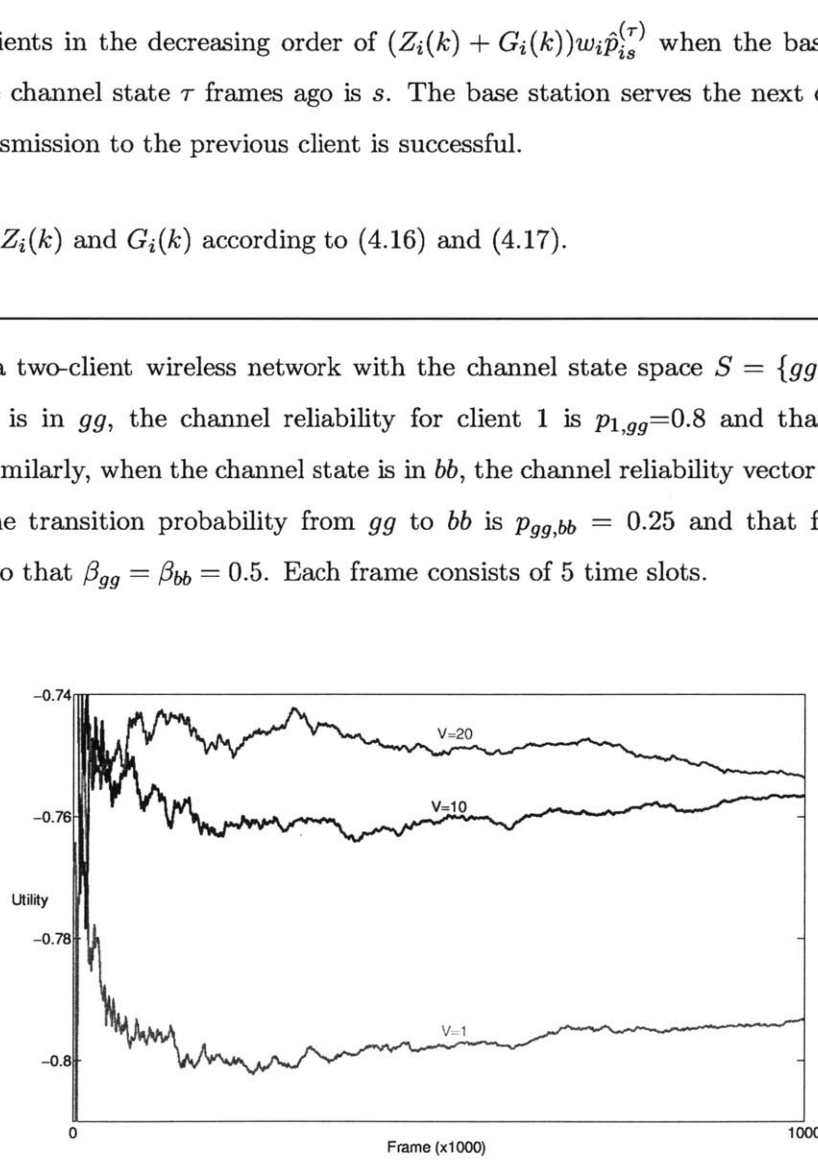



Figure 4-6: The utility optimal policy over time-varying channels with delayed channel state information

We apply the utility-optimal policy to solve a utility maximization problem below subject to throughput requirements:

$$\begin{aligned} \text{maximize} \quad & 2\log(d_1) + \log(d_2) \\ \text{subject to} \quad & (d_1, d_2) \in \Lambda, \quad d_1 \geq 0.7, \ \ d_2 \geq 0.7 \end{aligned}$$

The optimal solution is $(0.819, 0.7)$ and the optimal utility is -0.758. In the simulation result shown in Figure 4-6, we assume that the channel state information is delayed by 2 frames. As the value of $V$ increases, the total utility gets closer to the actual optimal value.

## 4.5 Summary

In this chapter, we studied a unicast network system with time-varying channels and formulated a utility maximization problem. The channel states change according to a Markov chain and we proved that the performance measure vector satisfies the strong conservation laws, so each vertex of the feasibility region is achieved by channel state dependent strict priority policies. The utility maximization problem is solved by following the drift-plus-penalty method.

In many real life wireless networks, the current channel state is difficult to obtain. Hence, we studied the effect of delayed channel state information as well. Delayed channel state information decreases the feasibility region. Because of the assumption that the channel states change according to a Markov chain, the knowledge of previous channel states enables the base station to estimate the current channel state. In this case, we can also show that the performance measure vector satisfies the strong conservation laws, so each vertex of the feasibility region is achieved by channel state dependent strict priority policies. The utility maximization problem is solved similarly by using the estimate of the current channel state instead.

Chapter 5

# Multicast with Instant Feedback

Some real-time data traffic is multicast in nature and should be treated differently from the unicast traffic. For example, in the real time broadcast system, different clients subscribe to the same flow and receive the same data packets. Unlike the unicast traffic, different clients that subscribe to the same flow may have different wireless channel qualities.

In [3], Hou and Kumar study multicast flows without feedback and develop a feasibility optimal policy by solving a maximization problem. Assuming no feedback is appropriate to analyze a large wireless network since instant feedback can be very difficult and impractical to obtain in such a large network. However, in a network of moderate size, instant feedback can be readily available and a scheduling policy can utilize such information. In this chapter, we focus on a wireless network of moderate size and assume that the instant feedback is always available.

We consider a multicast downlink network with clients subscribing to a set of flows from the base station. Each real time traffic has a delivery deadline of $T$ time slots, i.e. each packet must be transmitted within $T$ time slots or else it is useless. In order to model the hard deadline requirement, we group $T$ time slots into a frame, packets arrive at the beginning of the frame, and all the undelivered packets are dropped at the end of the frame. Therefore, any packets that are successfully delivered have delays less than $T$ time slots. This model can be used to analyze video applications that generate periodic flows of data packets. Every client has a delivery ratio requirement for each flow it subscribes to, i.e. the long-term ratio of packets that must meet the deadline.

We characterize the feasibility region, which is the set of all the delivery ratio requirement vectors that can be supported by any scheduling policy and develop a feasibility optimal policy, which can satisfy any delivery ratio requirement vectors that are within the feasibility region.

Our main contributions are

- We examine the effect of feedback on the feasibility region when there are multiple multicast flows in a wireless network
- We adopt and strengthen the analytic framework for addressing QoS constraints in wireless networks in [2].
- We devise a method to numerically characterize the feasibility region and apply it to the multicast system.
- We prove that the max-weight policy (similar to the LCQ policy defined in [12]), which maximizes the expected weighted sum throughput, is feasibility optimal.
- We prove that the greedy policy, which maximizes the expected immediate reward, is a max-weight policy in the multicast system with instant feedback.

The remainder of the chapter is organized as follows. We introduce the basic network model in Section 5.1. In Section 5.2, the multicast feasibility region is characterized by using the similar method. We examine how much feedback improves the feasibility region compared to no feedback in Section 5.3. We prove that the greedy policy is feasibility optimal in Section 5.4.

## 5.1 System Model

Consider a wireless system where there is one base station sending data flows with delay constraints to $N$ wireless clients. This model is similar to the model used in [3]. Throughout the chapter, $\mathbb{I}$ denotes the set of all the data flows in the system. A flow is subscribed to by a subset of clients, i.e. a subset of clients receive data packets from a flow. For convenience, we abuse the notation by defining $\mathcal{A} \in \mathbb{I}$ as the subset of clients who subscribe to data flow $\mathcal{A}$. For instance, consider a network with two flows and four clients in Figure 5-1. Client 1 and client 2 subscribe to flow 1, and client 3 and client 4 subscribe to flow 2. Then, $\mathbb{I} = \{\text{flow } 1, \text{flow } 2\}$, flow $1 = \{\text{client } 1, \text{client } 2\}$, and flow $2 = \{\text{client } 3, \text{client } 4\}$.

Since the link between the base station and a client is wireless, we assume that client $n$ has a channel reliability of $p_n$, i.e. when the base station transmits a packet to client $n$, the probability of successful transmission is $p_n$. For simplicity, we assume that the value of $p_n$ does not change over

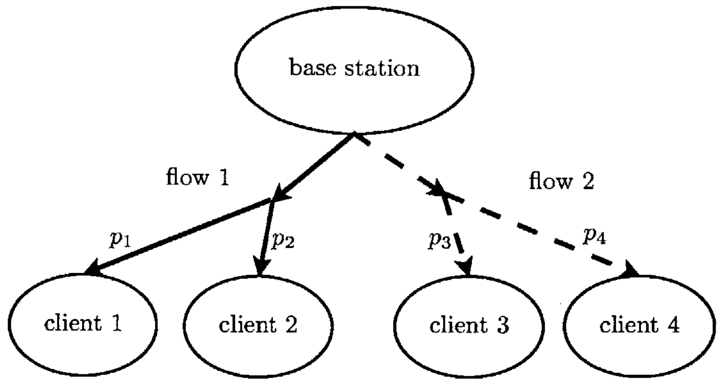



Figure 5-1: Example multicast network with two flows and four clients

time. However, the value of $p_n$ depends on the client and the different values reflect that wireless links can vary in quality from client to client as shown in Figure 5-1.

We consider a time-slotted system and group $T$ time slots into a frame as shown in Figure 2-1. Each traffic flow generates one packet at the beginning of the frame. The base station transmits a packet of a flow in each slot. Note that since the wireless channels are unreliable, not all packets may be delivered.

A time slot is long enough for a transmission of a packet to occur. Thus, if a packet transmission is scheduled during a time slot and is successful, then the packet is delivered by the end of the time slot. The base station receives ACK/NACKs from all clients by the end of the time slot during which a packet is served.

Define the delay of a packet as the number of time slots between the arrival of the packet to the base station and the end of the time slot when the successful transmission happens. For example, if packet 1 is successfully delivered during time slot 0 in Figure 2-1, then the delay of packet 1 is one time slot. To model the hard deadline constraints, we assure that the delay of each delivered packet is less than $T$ by dropping any undelivered packets at the end of a frame as shown in Figure 2-1.

We consider the class of work-conserving and non-anticipatory scheduling policies and denote the class by $\Pi$. The performance measure we are interested in is the long-term proportion of packets delivered. For a scheduling policy $\eta \in \Pi$, we define $D^{\eta}_{\mathcal{A},i}(k)$ to be the indicator random variable that is equal to 1 if client $i$ receives the packet from flow $\mathcal{A}$ during the interval $[kT, (k+1)T)$ by following policy $\eta$ and 0 otherwise. Then, the long-term proportion $\tilde{q}^{\eta}_{\mathcal{A},i}$ of packets under policy $\eta$

from flow $\mathcal{A}$ delivered to client $i$ can be written as:

$$\tilde{q}^{\eta}_{\mathcal{A},i} = \liminf_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} D^{\eta}_{\mathcal{A},i}(k)$$

Let $\mathcal{H}^{\eta}_{k-1}$ denote the history of all packet deliveries up to and including frame $k-1$ under policy $\eta$. That is, $\mathcal{H}^{\eta}_{k-1}$ is a vector $(\vec{D}^{\eta}(0), \cdots, \vec{D}^{\eta}(k-1))$ of indicator variable vectors $\vec{D}^{\eta}(t) = (D^{\eta}_{\mathcal{B},\ell}(t))_{\mathcal{B}\in\mathbb{I},\ell\in\{1,\cdots,N\}}$. Then, we define the expected long-term throughput $\hat{q}^{\eta}_{\mathcal{A},i}$ for client $i$ subscribing to flow $\mathcal{A}$ as

$$\begin{aligned}\hat{q}^{\eta}_{\mathcal{A},i} :&= \liminf_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} E[D^{\eta}_{\mathcal{A},i}(k)|\mathcal{H}^{\eta}_{k-1}] \\ &= \liminf_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} E[D^{\eta}_{\mathcal{A},i}(k)|\vec{D}^{\eta}(0), \cdots, \vec{D}^{\eta}(k-1)].\end{aligned}$$

By setting $b_k = k$ and applying Theorem 2.1.1, we can show that $\tilde{q}^{\eta}_{\mathcal{A},i} = \hat{q}^{\eta}_{\mathcal{A},i}$ with probability 1. As before, we use $\hat{q}^{\eta}_{\mathcal{A},i}$ exclusively to characterize the feasibility region and the feasibility optimal policy.

As part of the QoS constraints, each client $n$ has a specified delivery ratio requirement $q_{\mathcal{A},n}$ for each flow $\mathcal{A} \in \mathbb{I}$.

**Definition 5.1.1** Delivery ratio requirement vector $(q_{\mathcal{A},n})_{\mathcal{A}\in\mathbb{I},n\in\{1,\cdots,N\}}$ is said to be fulfilled by policy $\eta \in \Pi$ if $\hat{q}^{\eta}_{\mathcal{A},n} \geq q_{\mathcal{A},n}$ for every client $n$ and every flow $\mathcal{A}$ with probability 1.

## 5.2 Feasibility Region

The multicast system generalizes the unicast system in that flows are destined to more than one client. The base station serves up to one packet per time slot. For simplicity, we assume that one time slot is long enough for transmitting one packet of a flow.

Define $q_{\mathcal{A},n}$ as a delivery ratio requirement for flow $\mathcal{A} \in \mathbb{I}$ that client $n$ subscribes to. To simplify the analysis, however, it is sufficient to assume that one client subscribes to only one flow.

**Example 5.2.1** Let client 1 and client 2 receive both flow 1 and flow 2 as shown in Figure 5-2. By creating new fictitious clients for each flow, we can reduce the multiple- flow-per-client case into an equivalent system in which each client subscribes to only one flow. In Figure 5-2, client 1 is split into client 1-1 and client 1-2. Client 1-1 receives flow 1 and client 1-2 receives flow 2. They

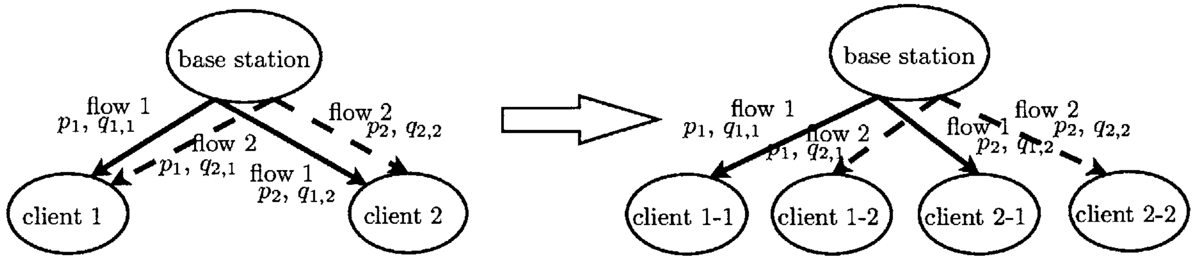



Figure 5-2: Multiple-flows-per-client system can be reduced to an equivalent system in which a client subscribes to one flow by creating fictitious clients; The fictitious clients have the same channel reliability as the original clients and the same delivery ratio requirements depending on which flow they receive.

both have the same channel reliability, $p_1$, as the original client. Because client 1-1 receives flow 1, its delivery ratio requirement is $q_{1,1}$. Similarly, client 1-2 has delivery ratio requirement $q_{2,1}$. Since every multicast system can be converted to another multicast system in which every client receives only one flow, hereinafter, we assume that a client subscribes to only one flow.

When a client receives only one flow, the client number is enough to index its delivery ratio requirement. Hereinafter, we use $q_i$ exclusively to denote the delivery ratio requirement for client $i$ for the flow it receives.

Let $\mathcal{A}(t) \in \mathbb{I}$ be the flow that the base station serves during time slot $t$ and $X_i(t)$ denote a random variable that is equal to 1 if a packet is successfully transmitted to client $i$ from its one and only flow during slot $t$ and 0 otherwise. Then, $U_{\mathcal{A}(t)}(t) := \{i | i \in \mathcal{A}(t), \sum_{\ell=0}^{t-1} X_i(\ell) = 0\}$ denotes the set of clients from flow $\mathcal{A}(t) \in \mathbb{I}$ who have not received packets yet at the beginning of time slot $t$. Because of our system model, any duplicate packets to a client are of no use, so $\sum_{i \in U_{\mathcal{A}(t)}} \alpha_i p_i$ is the expected immediate reward of serving flow $\mathcal{A}(t)$ during time slot $t$.

Given a weight vector $(\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, we consider the following greedy policy for the multicast system:

---

GREEDY POLICY FOR MULTICAST SYSTEM

(i) At the beginning of time slot $t$, calculate $r_{\mathcal{A}}(t) := \sum_{i \in U_{\mathcal{A}(t)}} \alpha_i p_i$ for $\forall \mathcal{A} \in \mathbb{I}$.

(ii) Serve the flow, $\mathcal{A}$, with the highest $r_{\mathcal{A}}(t)$.

---

By using the method of induction similar to the one we use to prove Theorem 2.2.1, we can indeed show that the above greedy policy maximizes the expected weighted sum throughput, i.e.

it achieves $\text{EWST}(\vec{\alpha})$. We can characterize the feasibility region by applying the greedy policy for different weight vectors as in the unicast case.

**Theorem 5.2.1** Consider a system in which there are multicast flows with instant feedback. Then, given a weight vector $\vec{\alpha} = (\alpha_i)_{i=1}^N$ such that $\alpha_i \geq 0$, the greedy policy, which serves flow $\mathcal{K}(t) := \arg\max_{\mathcal{A}\in\mathbb{I}} \sum_{i\in U_{\mathcal{A}}(t)} \alpha_i p_i$ at time slot $t$, achieves the maximum expected weighted sum throughput, $\text{EWST}(\vec{\alpha})$.

*Proof.* See Appendix B □

## 5.3 Feedback vs. No Feedback

To study how feedback affects the feasibility region in the multicast system, for different weight vectors ($\vec{\alpha}$), we compare $\text{EWST}(\vec{\alpha})$ under instant feedback, denoted by $\text{EWST}_{\text{fb}}(\vec{\alpha})$, and $\text{EWST}(\vec{\alpha})$ under no feedback, denoted by $\text{EWST}_{\text{nofb}}(\vec{\alpha})$, in Figure 5-3.

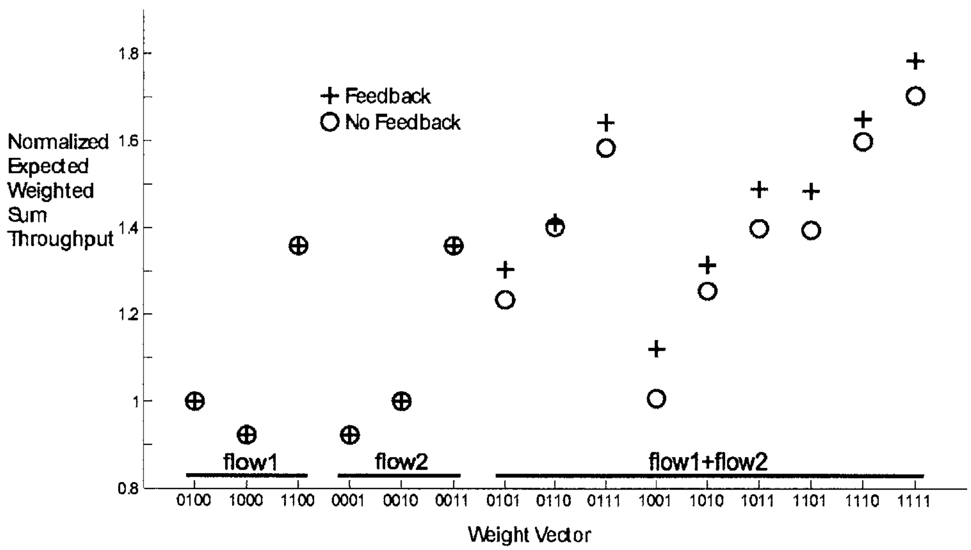



Figure 5-3: Normalized EWST is higher when there is instant feedback than when there is no feedback at all; $T = 5$, client 1 ($p_1 = 0.4$) and client 2 ($p_2 = 0.8$) belong to flow 1 and client 3 ($p_3 = 0.9$) and client 4 ($p_4 = 0.4$) belong to flow 2; $x$-axis represents a weight vector $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ and $y$-axis represents $\text{EWST}(\vec{\alpha})/\|\vec{\alpha}\|$.

In order to characterize the feasibility region for the no feedback case, we follow the scheme from [3]. Let $n_{\mathcal{A}}$ denote the number of time slots the base stations spend to serve flow $\mathcal{A}$. Then, the expected weighted sum throughput is given by $\sum_{\mathcal{A}\in\mathbb{I}} \sum_{i\in\mathcal{A}}(1-(1-p_i)^{n_{\mathcal{A}}})$. Whenever $\sum_{\mathcal{A}\in\mathbb{I}} n_{\mathcal{A}} < T$, the base station can serve any flow during the idling time slots and doing so does not decrease the expected weighted sum throughput. Hence, to find the maximum, we only consider the cases in which $\sum_{\mathcal{A}\in\mathbb{I}} n_{\mathcal{A}} = T$:

$$\mathrm{EWST_{nofb}}(\vec{\alpha}) = \max_{\sum_{\mathcal{A}\in\mathbb{I}} n_{\mathcal{A}}=T} \sum_{\mathcal{A}\in\mathbb{I}} \sum_{i\in\mathcal{A}} (1-(1-p_i)^{n_{\mathcal{A}}})$$

Hou and Kumar provide a simple algorithm to solve this maximization problem (Algorithm 1 in [3]).

As shown in Figure 5-3, $\mathrm{EWST_{fb}}(\vec{\alpha}) \geq \mathrm{EWST_{nofb}}(\vec{\alpha})$. Note that when there is only one flow in the system, i.e. the weight vector contains components associated with only one of the flows, $\mathrm{EWST_{fb}}(\vec{\alpha}) = \mathrm{EWST_{nofb}}(\vec{\alpha})$. This is because both with and without feedback the optimal policy will serve the one and only flow.

Next, we study by how much feedback increases $\mathrm{EWST}(\vec{\alpha})$ under different settings. First, we look at the effect of the number of clients per flow.

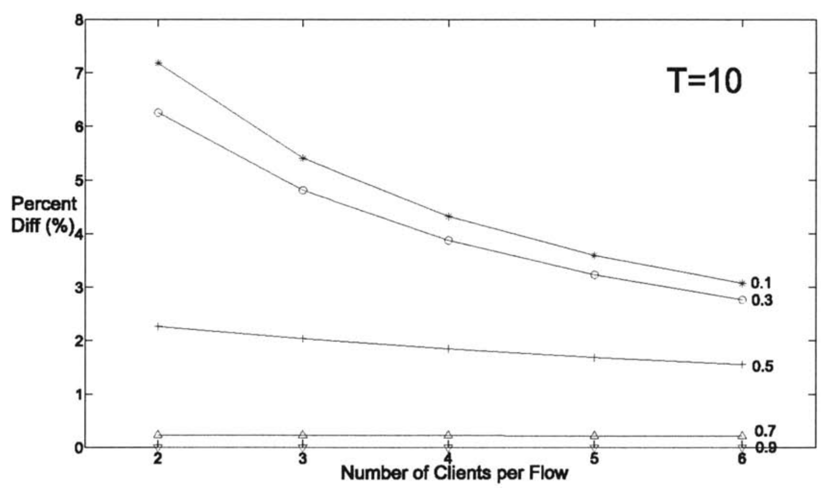



Figure 5-4: The percentage difference between $\mathrm{EWST_{fb}}$ and $\mathrm{EWST_{nofb}}$ (defined as $\frac{\mathrm{EWST_{fb}}-\mathrm{EWST_{nofb}}}{\mathrm{EWST_{fb}}} \times 100$) decreases as the number of clients per flow increases;we consider two flows to which the same number of clients subscribe; every client has the same channel reliability; the weight vector $\vec{\alpha} = \{1, 1, \cdots, 1\}$ is used; we consider channel reliability of 0.1, 0.3, 0.5, 0.7, and 0.9.

Figure 5-4 shows that the percentage difference decreases as the number of clients per flow increases. As the number of clients per flow increases, it becomes more and more unlikely that all the clients who subscribe to the same flow successfully receive the packet. Therefore, serving any flow blindly as in the no feedback case gives higher expected weighted sum throughput when the number of clients per flow is larger. For example, when there is only one client per flow whose channel reliability is $p$, with probability $p$ the client receives the packet successfully and serving the same flow in the next time slot results in no increase in the expected weighted sum throughput. However, when there are $n$ such clients per flow, the probability that serving the same flow in the next time slot results in no gain is $p^n$, the probability of all the $n$ clients receiving the packet successfully.

Next, we change the number of flows while keeping the number of clients per flow constant. Figure 5-5 shows that initially the difference increases although eventually converges to zero. The number of flows at which the maximum difference occurs increases as the channel reliability increases.

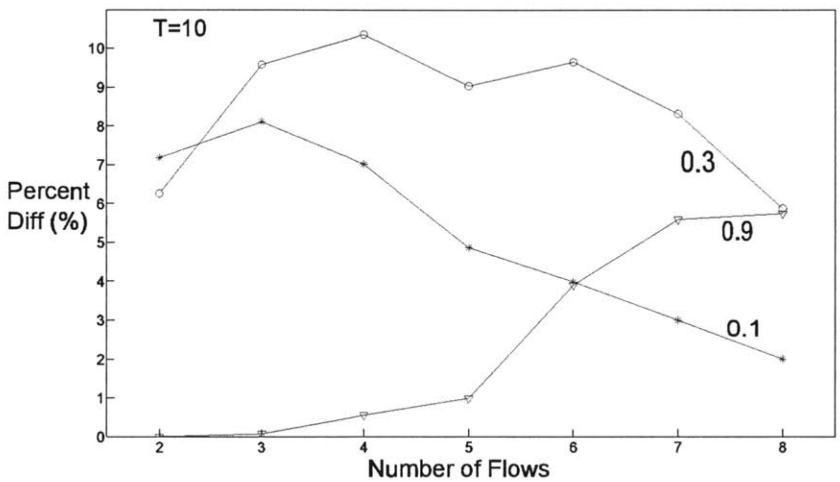



Figure 5-5: The difference increases as the number of flows increases and after a certain point it decreases;the point where the maximum difference occurs moves to the right as the channel reliability increases (the channel reliability for each client is marked on the right side) ; there are two clients per flow and every client has the same channel reliability

We define EWST per client to be $\frac{1}{N}\text{EWST}(\vec{\alpha})$, the maximum expected weighted sum throughput divided by the total number of clients. First, we fix the number of flows and the number of clients per flow and vary the number of time slots per frame in Figure 5-6. As the number of time slots per frame increases, EWST per client increases.

Next, we fix the number of time slots per frame and the number of clients per flow and vary the number of flows in Figure 5-7. As the number of flows increases, the EWST per client decreases because $\text{EWST}(\vec{\alpha})$ grows sublinearly with respect to the total number of clients.

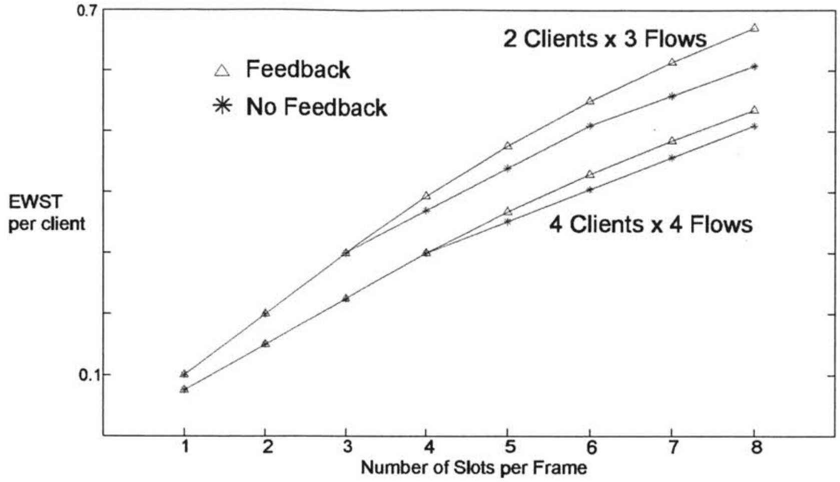



Figure 5-6: As the number of time slots per frame increases, EWST per client increases; every client has channel reliability $p = 0.3$ and each flow has the same number of clients;

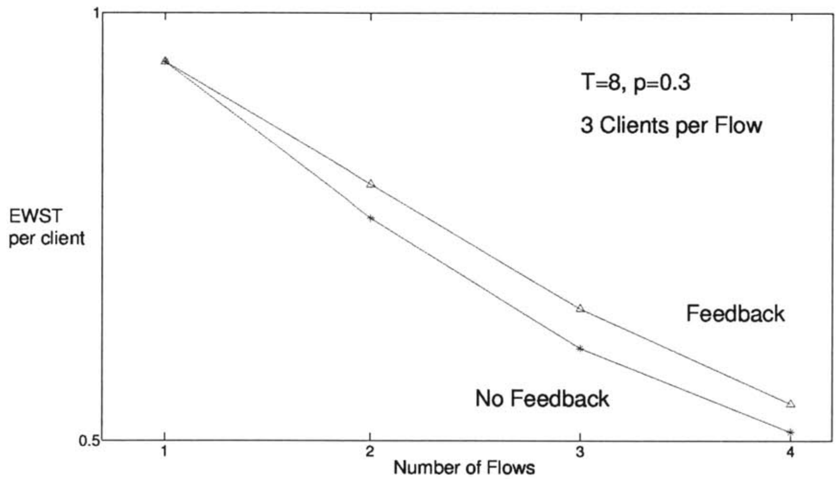



Figure 5-7: As the number of flow increases, the EWST per client decreases both under feedback and under no feedback; every client has channel reliability $p = 0.3$ and each flow has the same number of clients (3 in this case);

## 5.4 Feasibility Optimal Policy

After characterizing the feasibility region, we propose a feasibility optimal policy for the multicast system so that we can serve any set of clients that are inside the feasibility region. Let $D_i(k)$ be the indicator random variable that is equal to 1 if client $i$ receives the packet from the flow it subscribes to during the interval $[kT, (k+1)T)$ and 0 otherwise. Let $\hat{D}_i(k) := E[D_i(k)|\mathcal{H}_{k-1}]$, where $\mathcal{H}_{k-1}$

is the history of all packet deliveries up to and including frame $k-1$. Then, we can define the delivery debt $d_i(k) := \sum_{j=0}^{k-1}(q_i - \hat{D}_i(j))$ for each client $i$ at the beginning of time slot $kT$.

At the beginning of time slot $kT$, the base station calculates $Debt(k) = [d_i(k)^+]$, the vector consisting of all the delivery debts for each client $i$ up to $kT$. Then, we can consider a policy $\eta^0$ which maximizes $\sum_{i=1}^{N} d_i(k)^+ \cdot \hat{D}_i(k)$. We prove that $\eta^0$ is a feasibility optimal policy. Note that maximizing $\sum_{i=1}^{N} d_i(k)^+ \cdot \hat{D}_i(k)$ is equivalent to maximizing the expected weighted sum throughput with the weight vector equal to $\vec{\alpha} = (d_i(k)^+)_{i=1}^{N}$.

---

FRAME-BASED MAX-WEIGHT POLICY:

(i) At the beginning of frame $k$, calculate the expected delivery ratio debt vector $Debt(k) = [d_i(k)^+]$

(ii) Maximize $\sum_i d_i(k)^+ \cdot \hat{D}_i(k)$.

---

From Section 5.2, we know that the greedy policy achieves the maximum expected weighted sum throughput, EWST($\vec{\alpha}$). Therefore, the following algorithm is throughput optimal.

---

FRAME-BASED GREEDY POLICY:

(i) At the beginning of frame $k$, calculate the expected delivery ratio debt vector $Debt(k) = [d_i(k)^+]$

(ii) At time slot $t \in [kT, (k+1)T)$, calculate $r_{\mathcal{A}}(t) := \sum_{i \in U_{\mathcal{A}}(t)} d_i(k)^+ \cdot p_i$ for $\forall \mathcal{A} \in \mathbb{I}$.

(iii) Serve the flow, $\mathcal{A}$, with the highest $r_{\mathcal{A}}(t)$.

---

Since we assume that one client receives only one flow, we can use $q_n$ instead of $q_{\mathcal{A},n}$ without confusion. As in [2], we define the difference between the required delivery ratio and the achieved delivery ratio as a deadline miss ratio (DMR) function.

We use the DMR function to compare the greedy policy (feedback), the round-robin policy (feedback), and Algorithm 1 from [3] (no feedback) in Figure 5-8. The delivery ratio requirement vector in Table 5.1 is chosen to lie outside the no feedback feasiblity region but inside the instant feedback feasibility region. Hence, as shown in Figure 5-8, the DMR function goes to 0 under the greed policy but does not go to 0 under Algorithm 1 from [3].

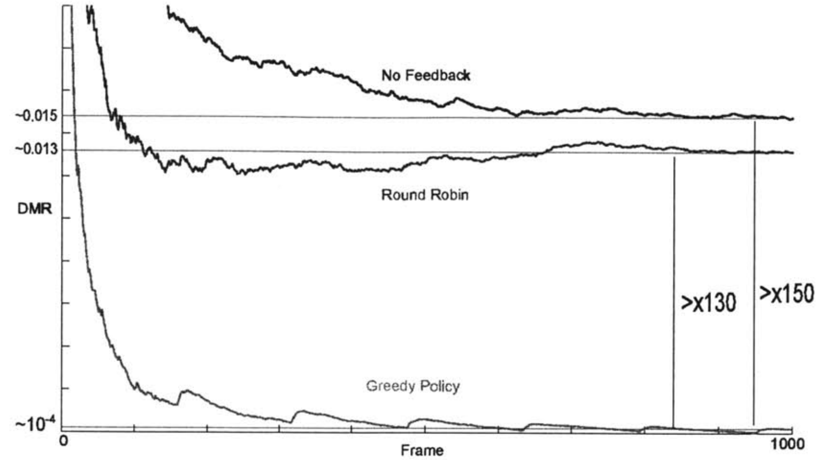



Figure 5-8: Greedy policy outperforms Round-robin and Algorithm 1 (labeled as No Feedback); $T = 5$, client 1 ($p_1 = 0.4$) and client 2 ($p_2 = 0.8$) subscribe to flow 1 and client 3 ($p_3 = 0.9$) and client 4 ($p_4 = 0.5$) subscribe to flow 2. The delivery ratio requirements are in Table 5.1.

Table 5.1: Delivery Ratio Requirements

| Flows | Clients | $q_i$'s |
|---|---|---|
| Flow 1 | Client 1 | 0.8327 |
| | Client 2 | 0.9964 |
| Flow 2 | Client 3 | 0.9968 |
| | Client 4 | 0.8232 |

The greedy policy outperforms the round-robin policy and Algorithm 1 from [3]. Also, the round-robin policy outperforms Algorithm 1 (No Feedback) because the round-robin utilizes feedback information. Because in Figure 5-8 the DMR function does not approach 0 under the round-robin policy, the round-robin policy is not feasibility optimal.

## 5.5 Summary

In this chapter, we studied a multicast network with instant feedback. Similar to the result in Chapter 2, the greedy policy is proved to be feasibility optimal. We compared the feasibility region of a multicast network with no feedback and that of a multicast network with instant feedback while varying the number of flows, the number of clients per flow, and the number of slots per frame. As the number of clients per flow increased, the difference decreased. As the number of flows increased, the difference decreased eventually, though not monotonically. As the number of slots per frame increased, the difference increased. In general, however, the percent difference was less than 10%. We suspect that the difference is not very significant because there are multiple clients per flow. When there are multiple clients per flow, it is unlikely that all clients receive packets and thereby serving any flow is likely to result in a positive increase in the expected weighted throughput.

Chapter 6

# Conclusions

In this thesis, we developed an alternative way to characterize the feasibility region, the set of achievable delivery ratio requirement vectors. In essence, the alternative numerical method is solving the maximization of the expected weighted sum throughput. By varying the weight vectors, we can characterize the outer boundary of the feasibility region. In addition, the alternative method turned out quite useful in characterizing feasibility optimal policies as well. In many situations, a max-weight policy is a feasibility optimal policy and the alternative method results in a max-weight policy with the debt vector as the weight vector.

We applied this method in a unicast system with delayed feedback in Chapter 3, in a unicast system with time-varying channels in Chapter 4, and in a multicast system with instant feedback in Chapter 5.

Other ways to expand the original model presented in [2] include heterogeneous delay bounds of packets, random delay of feedback signals, and aperiodic packet arrivals. These modifications of the model can be difficult to analyze because they alter the basic features of the model. For example, the heterogeneous delay bounds and aperiodic packet arrivals render the frame-based analysis difficult. Considering the versatility of the numerical method, however, we expect to be able to find at least a partial characterization of the feasibility regions.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

# Proof of Theorem 2.2.1

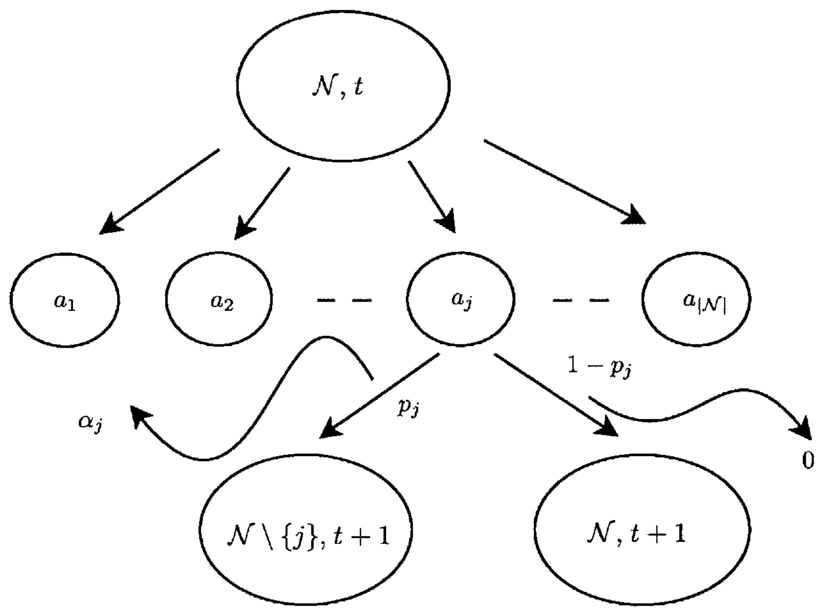



Figure A-1: Markov Decision Process; a state can be defined as the set of clients, $\mathcal{N}$, whose packets have not been delivered yet and the current time slot $t$. There are $|\mathcal{N}|$ number of actions to take at state $(\mathcal{N}, t)$; each action represents the client to serve during time slot $t$. If the base station takes action $a_j$, with probability $p_j$ the system gets reward $\alpha_j$ and client $j$ has been served successfully. Otherwise, the reward is 0.

The system can be modeled as a Markov decision process shown in Figure A-1. Let $S$ denote the set of all states. Then, $s = (\mathcal{N}, t) \in S$ where $\mathcal{N} \subset \{1, 2, \cdots, N\}$ and $t \in \{0, 1, 2, \cdots, T\}$. Here, $\mathcal{N}$ represents the set of clients who have not received packets until the beginning of the current time slot and $t$ represents the current time slot.

Given $s = (\mathcal{N}, t)$ and $\mathcal{N} \neq \emptyset$, the set of actions available from state $s$ is $A_s := \{a_1, \cdots, a_{|\mathcal{N}|}\}$ in which $a_j$ represents serving the $j$th client in $\mathcal{N}$. Therefore, when $t \leq T - 1$, taking action $a_j$ from state $s$ leads to two possible states: $(\mathcal{N}, t+1)$ if the packet transmission fails and $(\mathcal{N} \setminus \{j\}, t+1)$ if the packet transmission succeeds.

Since we are maximizing the expected weighted sum throughput, given a weight vector $(\alpha_i)_{i=1}^N$, when $\mathcal{N} \neq \emptyset$, the reward of $\alpha_j$ is acquired when the packet transmission to client $j$ is successful.

Similar to [1], we define a policy as a sequence of functions $\pi = \{\mu_0, \cdots, \mu_{T-1}\}$. $\mu_t$ maps states $(\mathcal{N}, t)$ for any $\mathcal{N} \subset \{1, \cdots, N\}$ into clients to serve at time slot $t$, i.e. $i(t) = \mu_t(\mathcal{N}, t) \in \mathcal{N}$. Here, the class of work-conserving and non-anticipatory scheduling policies is the class of admissible policies, denoted by $\Pi$.

Let $J_\pi(s_0)$ be the expected reward of $\pi$ starting at $s_0 := (\{1, \cdots, N\}, 0)$. Then, because of our reward definition, $J_\pi(s_0)$ is equal to $\sum_{i=1}^N E[\alpha_i D_i^\pi(0)]$, the expected weighted sum throughput under policy $\pi$. We want to show that the greedy policy, which maximizes the expected immediate reward, maximizes the total expected reward, i.e. the expected weighted sum throughput.

According to Proposition 1.3.1 in [1], the optimal cost $J^*(s_0)$ is equal to $J_0(s_0)$, given by the last step of the following algorithm, which proceeds backward in time from $T-1$ to 0:

$$J_T(\mathcal{N}, T) = 0$$

$$J_t(\mathcal{N}, t) = \begin{cases} \max_{i \in \mathcal{N}} [\alpha_i p_i + p_i J_{t+1}(\mathcal{N} \setminus \{i\}, t+1) + (1-p_i) J_{t+1}(\mathcal{N}, t+1)] & \text{if } \mathcal{N} \neq \emptyset; \\ 0 & \text{if } \mathcal{N} = \emptyset. \end{cases}$$

Furthermore, if $\mu_t^*(\mathcal{N}, t)$ satisfies:

$$\mu_t^*(\mathcal{N}, t) = \arg\max_{i \in \mathcal{N}} [\alpha_i p_i + p_i J_{t+1}(\mathcal{N} \setminus \{i\}, t+1) + (1-p_i) J_{t+1}(\mathcal{N}, t+1)]$$

when $\mathcal{N} \neq \emptyset$, then the policy $\pi^* = \{\mu_0^*, \cdots, \mu_{T-1}^*\}$ is optimal.

In order to show that the greedy policy maximizes the expected weighted sum throughput, we show that for $0 \leq t \leq T-1$

$$\mu_t^*(\mathcal{N}, t) = \arg\max_{i \in \mathcal{N}} \alpha_i p_i \tag{A.1}$$

When $|\mathcal{N}| = 1$, (A.1) is always true since there is only one client left and all the admissible policies are non-idling. When $\mathcal{N} = \emptyset$, the base station is forced to be idle because all the clients have been served. Thus, hereinafter, we assume that $|\mathcal{N}| \geq 2$.

At time $T-1$, $J_{T-1}(\mathcal{N}, T-1) = \max_{i\in\mathcal{N}} \alpha_i p_i$. Thus, $\mu^*(\mathcal{N}, T-1) = \arg\max_{i\in\mathcal{N}} \alpha_i p_i$.

Consider the subproblem whereby the system is at $(\mathcal{N}, T-2)$ and we wish to maximize the expected reward from $T-2$ to $T$. Consider two policies $\pi_1 = \{\mu^{(1)}_{T-2}, \mu^*_{T-1}\}$ and $\pi_2 = \{\mu^{(2)}_{T-2}, \mu^*_{T-1}\}$ such that

$$
\begin{aligned}
\mu^{(1)}_{T-2}(\mathcal{N}, T-2) &= k_1 := \arg\max_{i\in\mathcal{N}} \alpha_i p_i \\
\mu^{(2)}_{T-2}(\mathcal{N}, T-2) &= \ell \neq k_1;
\end{aligned}
$$

Let $k_2 = \arg\max_{i\in\mathcal{N}\setminus\{k_1\}} \alpha_i p_i$. Then,

$$
\begin{aligned}
J_{\pi_1}(\mathcal{N}, T-2) &= \alpha_{k_1} p_{k_1} + p_{k_1}\alpha_{k_2} p_{k_2} + (1-p_{k_1})\alpha_{k_1} p_{k_1} \\
J_{\pi_2}(\mathcal{N}, T-2) &= \alpha_\ell p_\ell + p_\ell \alpha_{k_1} p_{k_1} + (1-p_\ell)\alpha_{k_1} p_{k_1} = \alpha_\ell p_\ell + \alpha_{k_1} p_{k_1}.
\end{aligned}
$$

By assumption, $\alpha_{k_1} p_{k_1} \geq \alpha_{k_2} p_{k_2} \geq \alpha_\ell p_\ell$. Therefore, $J_{\pi_1}(\mathcal{N}, T-2) \geq J_{\pi_2}(\mathcal{N}, T-2)$ for any $\ell \neq \arg\max_{i\in\mathcal{N}} \alpha_i p_i$ and $\mu^*(\mathcal{N}, T-2) = \arg\max_{i\in\mathcal{N}} \alpha_i p_i$.

To prove by backward induction, given $0 < t < T$, assume that for $\forall k \geq t$, the optimal policy at $(\mathcal{N}, k)$ is $\mu^*(\mathcal{N}, k) = \arg\max_{i\in\mathcal{N}} \alpha_i p_i$.

As before, we consider the subproblem whereby the system is at $(\mathcal{N}, t-1)$ and we wish to maximize the expected reward from $t-1$ to $T$. Consider two policies $\pi_1 = \{\mu^{(1)}_{t-1}, \mu^*_t, \cdots, \mu^*_{T-1}\}$ and $\pi_2 = \{\mu^{(2)}_{t-1}, \mu^*_t, \cdots, \mu^*_{T-1}\}$ such that

$$
\begin{aligned}
\mu^{(1)}_{t-1}(\mathcal{N}, t-1) &= k_1 := \arg\max_{i\in\mathcal{N}} \alpha_i p_i \\
\mu^{(2)}_{t-1}(\mathcal{N}, t-1) &= \ell \neq k_1;
\end{aligned}
$$

Let $\pi_3$ denote the third policy which serves $k_1$ in time slot $t-1$ and $\ell$ in time slot $t$. In other words, $\pi_3 = \{\mu^{(3)}_{t-1}, \mu^{(3)}_t, \mu^*_{t+1}, \cdots, \mu^*_{T-1}\}$ such that

$$
\begin{aligned}
\mu^{(3)}_{t-1}(\mathcal{N}, t-1) &= k_1 \\
\mu^{(3)}_t(\mathcal{N}, t) &= \mu^{(3)}_t(\mathcal{N}\setminus\{k_1\}, t) = \ell;
\end{aligned}
$$

By assumption, the truncated policy $\{\mu^{(3)}_t, \mu^*_{t+1}, \cdots, \mu^*_{T-1}\}$ is not optimal. Therefore, because of Principle of Optimality, $J_{\pi_1}(\mathcal{N}, t-1) \geq J_{\pi_3}(\mathcal{N}, t-1)$.

By definition, $J_{\pi_3}(\mathcal{N}, t-1)$ can be written as:

$$\begin{aligned}
J_{\pi_3}(\mathcal{N}, t-1) &= \alpha_{k_1}p_{k_1} + p_{k_1}J_{\pi_3}(\mathcal{N} \setminus \{k_1\}, t) + (1-p_{k_1})J_{\pi_3}(\mathcal{N}, t) \\
&= \alpha_{k_1}p_{k_1} + p_{k_1}(\alpha_\ell p_\ell + p_\ell J^*(\mathcal{N} \setminus \{k_1, \ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N} \setminus \{k_1\}, t+1)) \\
&+ (1-p_{k_1})(\alpha_\ell p_\ell + p_\ell J^*(\mathcal{N} \setminus \{\ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N}, t+1)) \\
&= \alpha_{k_1}p_{k_1} + \alpha_\ell p_\ell + p_{k_1}(p_\ell J^*(\mathcal{N} \setminus \{k_1, \ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N} \setminus \{k_1\}, t+1)) \\
&+ (1-p_{k_1})(p_\ell J^*(\mathcal{N} \setminus \{\ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N}, t+1)). \qquad \text{(A.2)}
\end{aligned}$$

Similarly, $J_{\pi_2}(\mathcal{N}, t-1)$ can be written as:

$$\begin{aligned}
J_{\pi_2}(\mathcal{N}, t-1) &= \alpha_\ell p_\ell + p_\ell J^*(\mathcal{N} \setminus \{\ell\}, t) + (1-p_\ell)J^*(\mathcal{N}, t) \\
&= \alpha_\ell p_\ell + p_\ell(\alpha_{k_1}p_{k_1} + p_{k_1}J^*(\mathcal{N} \setminus \{k_1, \ell\}, t+1) + (1-p_{k_1})J^*(\mathcal{N} \setminus \{\ell\}, t+1)) \\
&+ (1-p_\ell)(\alpha_{k_1}p_{k_1} + p_{k_1}J^*(\mathcal{N} \setminus \{k_1\}, t+1) + (1-p_{k_1})J^*(\mathcal{N}, t+1)) \\
&= \alpha_{k_1}p_{k_1} + \alpha_\ell p_\ell + p_{k_1}(p_\ell J^*(\mathcal{N} \setminus \{k_1, \ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N} \setminus \{k_1\}, t+1)) \\
&+ (1-p_{k_1})(p_\ell J^*(\mathcal{N} \setminus \{\ell\}, t+1) + (1-p_\ell)J^*(\mathcal{N}, t+1)). \qquad \text{(A.3)}
\end{aligned}$$

The second equality follows because $k_1 = \arg\max_{i \in \mathcal{N}} \alpha_i p_i \Rightarrow k_1 = \arg\max_{i \in \mathcal{N} \setminus \{\ell\}} \alpha_i p_i$ when $\ell \neq k_1$ and we assume that $\mu^*(\mathcal{N}, k) = \arg\max_{i \in \mathcal{N}} \alpha_i p_i$ for $\forall k \geq t$ and $\mathcal{N} \subset \{1, \cdots, N\}$.

Since (A.2) and (A.3) are the same, $J_{\pi_1}(\mathcal{N}, t-1) \geq J_{\pi_3}(\mathcal{N}, t-1) = J_{\pi_2}(\mathcal{N}, t-1)$. The inequality holds for any $\ell \neq k_1$, so $\mu^*(\mathcal{N}, t-1) = \arg\max_{i \in \mathcal{N}} \alpha_i p_i$.

In conclusion, the greedy policy which serves client $\arg\max_{i \in \mathcal{N}} \alpha_i p_i$ maximizes the expected weighted sum throughput where $\mathcal{N}$ is the set of clients that have not received packets successfully.

Appendix B

# Proof of Theorem 5.2.1

We use the method of backward induction similar to the one we use to prove Theorem 2.2.1 in Appendix A.

As in [1], a policy consists of a sequence of functions $\pi = \{\mu_0, \cdots, \mu_{T-1}\}$. where $\mu_t$ maps set $\mathcal{N}(t)$ of clients who have not received packets yet by the beginning of time slot $t$ into flow $\mathcal{A}(t)$ to serve during time slot $t$. For example, the greedy policy consists of a sequence of functions such that $\mu_t^{\text{greedy}}(\mathcal{N}(t)) = \arg\max_{\mathcal{A}\in\mathbb{I}} \sum_{i\in U_{\mathcal{A}}(t)} \alpha_i p_i$.

At the end of a frame, i.e. at time slot $T-1$, the base station should serve flow $\mathcal{K}(T-1) := \arg\max_{\mathcal{A}\in\mathbb{I}} \sum_{i\in U_{\mathcal{A}}(T-1)} \alpha_i p_i$ to maximize the expected weighted sum throughput. For any policy $\pi = \{\mu_0, \cdots, \mu_{T-1}\}$, we can define $\pi' = \{\mu_0, \cdots, \mu_{T-2}, \mu_{T-1}^{\text{greedy}}\}$. Then,

$$\sum_{i=1}^{N} E[\alpha_i D_i^{\pi}(0)] \leq \sum_{i=1}^{N} E[\alpha_i D_i^{\pi'}(0)].$$

Therefore, applying the greedy policy at $T-1$ is optimal at $T-1$.

Given $t$, assume that applying the greedy policy from time slot $t$ and onwards is optimal. That is, for any policy $\pi = \{\mu_0, \cdots, \mu_{T-1}\}$, we define $\pi' := \{\mu_0, \cdots, \mu_{t-1}, \mu_t^{\text{greedy}}, \cdots, \mu_{T-1}^{\text{greedy}}\}$ and assume that

$$\sum_{i=1}^{N} E[\alpha_i D_i^{\pi}(0)] \leq \sum_{i=1}^{N} E[\alpha_i D_i^{\pi'}(0)]. \tag{B.1}$$

Then, we want to show that at time slot $t-1$, serving flow $\mathcal{K}(t-1) = \arg\max_{\mathcal{A}\in\mathbb{I}} \sum_{i\in U_{\mathcal{A}}(t-1)} \alpha_i p_i$ leads to the expected weighted sum throughput larger than or equal to that obtained by $\pi'$. That is, if we define $\pi'' := \{\mu_0, \cdots, \mu_{t-2}, \mu_{t-1}^{\text{greedy}}, \cdots, \mu_{T-1}^{\text{greedy}}\}$, then we want to show that

$$\sum_{i=1}^{N} E[\alpha_i D_i^{\pi'}(0)] \leq \sum_{i=1}^{N} E[\alpha_i D_i^{\pi''}(0)].$$

Given a subset $\mathcal{N}(t-1)$ of clients who have not received packets yet at the beginning of time slot $t-1$, we assume that $\mu_{t-1}(\mathcal{N}(t-1)) \neq \mu_{t-1}^{\text{greedy}}(\mathcal{N}(t-1))$ because otherwise following the truncated policy $(\pi')_{t-1}^{T-1}$ from $\mathcal{N}(t-1)$ results in the same expected weighted sum throughput as following the truncated policy $(\pi'')_{t-1}^{T-1}$ from $\mathcal{N}(t-1)$.

Consider a policy $\pi_1 := \{\mu_0, \cdots, \mu_{t-2}, \mu_{t-1}^{\text{greedy}}, \mu_t = \mu_{t-1}, \mu_{t+1}^{\text{greedy}}, \cdots, \mu_{T-1}^{\text{greedy}}\}$ which serves the same flows as $\pi$ until time slot $t-2$, serves the flow that maximizes the expected immediate reward at time slot $t-1$, serves the flow that $\pi$ would have served at time slot $t-1$ at time slot $t$, and from time slot $t+1$ serves the flow that maximizes the expected immediate reward.

We define $\mathbb{S} := \{s \subset \{1, \cdots, N\} : P\{\mathcal{N}(t-1) \to s | \mu_{t-1}(\mathcal{N}(t-1))\} > 0\}$ to be the set of subsets which have positive transition probability from $\mathcal{N}(t-1)$ given flow $\mu_{t-1}(\mathcal{N}(t-1))$ is served during time slot $t-1$. Then, because $\mu_{t-1}(\mathcal{N}(t-1)) \neq \mu_{t-1}^{\text{greedy}}(\mathcal{N}(t-1))$, for any $s \in \mathbb{S}$,

$$\mu_{t-1}^{\text{greedy}}(\mathcal{N}(t-1)) = \mu_t^{\text{greedy}}(s).$$

This implies that following the truncated policy

$(\pi_1)_{t-1}^{T-1} = \{\mu_{t-1}^{\text{greedy}}, \mu_t = \mu_{t-1}, \mu_{t+1}^{\text{greedy}}, \cdots, \mu_{T-1}^{\text{greedy}}\}$ from $\mathcal{N}(t-1)$ results in the same expected weighted sum throughput as following the truncated policy

$(\pi')_{t-1}^{T-1} = \{\mu_{t-1}, \mu_t^{\text{greedy}}, \cdots, \mu_{T-1}^{\text{greedy}}\}$ from $\mathcal{N}(t-1)$.

Because $\pi_1' = \pi''$, the assumption expressed in (B.1) implies that the expected weighted sum throughput obtained by following the truncated policy $(\pi'')_{t-1}^{T-1}$ is greater than or equal to that obtained by $(\pi_1)_{t-1}^{T-1}$. Therefore, for any subset $\mathcal{N}(t-1)$, the expected weighted sum throughput obtained by following the truncated policy $(\pi'')_{t-1}^{T-1}$ is greater than or equal to that obtained by $(\pi')_{t-1}^{T-1}$ and thereby the following holds:

$$\sum_{i=1}^{N} E[\alpha_i D_i^{\pi'}(0)] \leq \sum_{i=1}^{N} E[\alpha_i D_i^{\pi''}(0)].$$

By the method of backward induction, we conclude that the greedy policy achieves the maximum expected weighted sum throughput, $\text{EWST}(\vec{\alpha})$.

Appendix C

# Proof of Theorem 2.3.1

Let's define the Lyapunov function for frame $k$:

$$L(Debt(k)) = \frac{1}{2}\sum_{i=1}^{N}(d_i(k)^+)^2. \tag{C.1}$$

Note that

$$\begin{aligned} d_i(k+1)^+ = (d_i(k) + q_i - \hat{D}_i(k))^+ &\leq (d_i(k)^+ + q_i - \hat{D}_i(k))^+ \\ &\leq (d_i(k)^+ - \hat{D}_i(k))^+ + q_i \end{aligned}$$

The first inequality follows because $d_i(k) \leq d_i(k)^+$ and the second inequality follows because $q_i \geq 0$. Note that the last equation resembles the update equation for a single-server discrete time queueing system if we define $Q_i(k) := d_i(k)^+$. However, if the expected debt were actually a single-server discrete time queue, we would have obtained an equality instead of the last inequality above so that $Q_i(k+1) = (Q_i(k) - \hat{D}_i(k))^+ + q_i$.

Using the approach described in [9], we obtain the following.

$$\begin{aligned}
L(Debt(k+1)) - L(Debt(k)) &= \frac{1}{2}\sum_{i=1}^{N}[(d_i(k+1)^+)^2 - (d_i^+(k))^2] \\
&\leq \frac{1}{2}\sum_{i=1}^{N}[((d_i(k)^+ - \hat{D}_i(k))^+ + q_i)^2 - (d_i^+(k))^2] \\
&\leq \frac{1}{2}\sum_{i=1}^{N}[(d_i(k)^+ - \hat{D}_i(k))^2 + 2q_i(d_i(k)^+ - \hat{D}_i(k))^+ + q_i^2 - (d_i^+(k))^2] \\
&= \frac{1}{2}\sum_{i=1}^{N}[-2d_i(k)^+\hat{D}_i(k) + (\hat{D}_i(k))^2 + 2q_i(d_i(k)^+ - \hat{D}_i(k))^+ + q_i^2] \\
&\leq \sum_{i=1}^{N} d_i(k)^+(q_i - \hat{D}_i(k)) + \frac{1}{2}\sum_{i=1}^{N}[q_i^2 + (\hat{D}_i(k))^2] \\
&\leq \sum_{i=1}^{N} d_i(k)^+(q_i - \hat{D}_i(k)) + B. \qquad \text{(C.2)}
\end{aligned}$$

for some constant $B$ since $q_i^2 + (\hat{D}_i(k))^2 \leq 2$.

Now define $\Delta(Debt(k))$ the conditional Lyapunov drift for frame $k$:

$$\Delta(Debt(k)) = E[L(Debt(k+1)) - L(Debt(k))|Debt(k)]. \qquad \text{(C.3)}$$

From (C.2), we have that $\Delta(Debt(k))$ for our case satisfies:

$$\Delta(Debt(k)) \leq B + \sum_{i=1}^{N} d_i(k)^+ q_i - \sum_{i=1}^{N} E[d_i(k)^+\hat{D}_i(k)|Debt(k)] \qquad \text{(C.4)}$$

Let $\eta^*(k)$ be a policy for frame $k$ which maximizes $\sum_{i=1}^{N} E[d_i(k)^+\hat{D}_i(k)|Debt(k)]$. Then, for any other policy, $\eta(k)$, we have:

$$\sum_{i=1}^{N} E[d_i(k)^+\hat{D}_i(\eta^*(k))|Debt(k)] \geq \sum_{i=1}^{N} E[d_i(k)^+\hat{D}_i(\eta(k))|Debt(k)] \qquad \text{(C.5)}$$

Plugging in (C.5) into (C.4), we obtain:

$$\Delta(Debt(k))^* \leq B + \sum_{i=1}^{N} d_i(k)^+ q_i - \sum_{i=1}^{N} E[d_i(k)^+\hat{D}_i(\eta(k))|Debt(k)] \qquad \text{(C.6)}$$

for any policy $\eta(k)$ for frame $k$. Here, the left-hand side is the conditional Lyapunov drift for frame

$k$ when we employ $\eta^*(k)$, the max-weight policy, during frame $k$. Rearranging the right-hand side of (C.6), we obtain:

$$\Delta(Debt(k))^* \leq B - \sum_{i=1}^{N} d_i(k)^+(E[\hat{D}_i(\eta(k))|Debt(k)] - q_i) \tag{C.7}$$

Let $(q_i) \in \Lambda - \epsilon\mathbf{1}$ where $\Lambda$ is the feasible region. Since there exists a stationary randomized policy that achieves the throughput vectors on the boundary of $\Lambda$, we can find $\eta$ such that

$$E[\hat{D}_i(\eta(k))|Debt(k)] = E[\hat{D}_i(\eta(k))] \geq q_i + \epsilon. \tag{C.8}$$

Plugging in (C.8) into (C.7), we obtain:

$$\Delta(Debt(k))^* \leq B - \epsilon \sum_{i=1}^{N} d_i(k)^+ \tag{C.9}$$

Taking the expection of (C.9) over the randomness of $Debt(k)$ values yields:

$$E[\Delta(Debt(k))^*] \leq B - \epsilon \sum_{i=1}^{N} E[d_i(k)^+] \tag{C.10}$$

Using the definition of $\Delta(Debt(k))$ in (C.3) and the law of iterated expectation yields:

$$E[\Delta(Debt(k))] = E[E[L(Debt(k+1)) - L(Debt(k))|Debt(k)]] = E[L(Debt(k+1))] - E[L(Debt(k))] \tag{C.11}$$

Substituting (C.11) into (C.10) yields:

$$E[L(Debt(k+1))^*] - E[L(Debt(k))^*] \leq B - \epsilon \sum_{i=1}^{N} E[d_i(k)^+] \tag{C.12}$$

Since (C.12) holds for all $k = \{0, 1, 2, \cdots\}$, summing over $k \in \{0, 1, \cdots, K-1\}$ for some integer $K > 0$ yields (by telescoping sums):

$$E[L(Debt(K))^*] - E[L(D(0))^*] \leq BK - \epsilon \sum_{k=0}^{K-1} \sum_{i=1}^{N} E[d_i(k)^+] \tag{C.13}$$

Therefore, by using the fact that $E[L(Debt(K))^*] \geq 0$, we obtain:

$$\frac{1}{K}\sum_{k=0}^{K-1}\sum_{i=1}^{N} E[d_i(k)^+] \leq \frac{B}{\epsilon} + \frac{1}{\epsilon K}E[L(D(0))^*] \tag{C.14}$$

Let $E[L(D(0))^*] < \infty$. Taking the limsup on both sides as $K$ goes to $\infty$ yields:

$$\limsup_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1}\sum_{i=1}^{N} E[d_i(k)^+] \leq \frac{B}{\epsilon} \tag{C.15}$$

Note that (C.15) implies that $d_i(k)^+$ is strongly stable for any $i$. Therefore, according to Theorem 2.8 in [9], $d_i(k)^+$ is rate stable for any $i$ so that $\limsup_{k\to\infty} \frac{d_i(k)^+}{k} = 0$ with probability 1.

Hou and Kumar prove the following lemma in [3]:

**Lemma C.0.1** A system is fulfilled by a policy $\eta$ if, under $\eta$, $\limsup_{k\to\infty} \frac{(d_i(k))^+}{k} = 0$ for all $i$ where $x^+ := \max\{x, 0\}$.

From Lemma C.0.1, we conclude that $(q_i)$ is indeed fulfilled. Since the max-weight policy proposed ensures that for any delivery requirement vector $(q_i) \in \Lambda - \epsilon\mathbf{1}$ inside the feasible region $\Lambda$ is achieved, the max-weight policy is indeed feasibility optimal.

# Bibliography


[1] Dimitri P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific Belmont, 1995.

[2] I.-H. Hou, V. Borkar, and P.R. Kumar. A theory of qos for wireless. In *INFOCOM 2009, IEEE*, pages 486 –494, april 2009.

[3] I-Hong Hou and P. R. Kumar. Broadcasting delay-constrained traffic over unreliable wireless links with network coding. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 4:1–4:10, New York, NY, USA, 2011. ACM.

[4] I-Hong Hou and P.R. Kumar. Scheduling heterogeneous real-time traffic over fading wireless channels. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[5] I-Hong Hou, PR Kumar, et al. Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 175–184. ACM, 2009.

[6] I.H. Hou and PR Kumar. Utility maximization for delay constrained qos in wireless. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[7] K. Jagannathan, S. Mannor, I. Menache, and E. Modiano. A state action frequency approach to throughput maximization over uncertain wireless channels. In *INFOCOM, 2011 Proceedings IEEE*, pages 491–495. IEEE, 2011.

[8] M Loeve. Probability theory, vols. 1–2, 1955.

[9] M.J. Neely. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

[10] R. Schneider. *Convex bodies: the Brunn-Minkowski theory*, volume 44. Cambridge University Press, 1993.

[11] J.G. Shanthikumar and D.D. Yao. Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Operations Research*, 40(3-Supplement-2):S293–S299, 1992.

[12] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *Information Theory, IEEE Transactions on*, 39(2):466–478, 1993.

[13] D. Yao. Dynamic scheduling via polymatroid optimization. *Performance Evaluation of Complex Systems: Techniques and Tools*, pages 89–113, 2002.